CALCOLATRICE ELETTRONICA
MODELLO ZERO

DESCRIZIONE FUNZIONALE
PARTE SECONDA (LOGICA)

Ing. C. Olivetti & C. Ivrea

C A L C O L A T R I C E E L E T T R O N I C A

O L I V E T T I

D E S C R I Z I O N E F U N Z I O N A L E

Parte 2ª

Pisa, 7 marzo 1957

I N D I C E

pag.

1 L A M E M O R I A 2

1.1 Generalità - Schema logico elementare 2

1.11. Definizione 2

1.12. Capacità della Memoria 2

1.13. Ciclo fondamentale di funzionamento della M. 3

1.2 Principi fisici di funzionamento della Memoria 5

1.2.1 Elemento fisico fondamentale di Memoria. 5

1.2.2 Principio di autocoincidenza nella selezione della Memoria 7

1.2.3. Principio di selezione mediante sottomatrici 8

1.2.4. Principio di lettura in una sottomatrice mediante descriminazione temporale. 9

1.3. Schema funzionale a blocchi della Memoria. 11

1.3.1. Introduzione 11

1.3.2. Matrici di nuclei magnetici 12

1.3.3. Selezione e pilotaggio della Memoria 13

1.3.4. Lettura della memoria 17

1.3.5. Inibizione e scrittura della Memoria 19

1.3.6. Forme d'onda di temporizzazione 20

2 L'ACCUMULATORE E I REGISTRI DI MODIFICAZIONE 21

2.1. Generalità e schema logico elementare 21

2.1.1. Definizione di A.- T. 21

2.1.2. Capacità di A.- T. 21

2.1.3. Ciclo fondamentale di funzionamento di A. - T. 22

2.2. Schema funzionale dell' A. - T. 24

2.2.1. Introduzione 24
2.2.2. Composizione 24
2.2.3. A) Organi di decodificazione, selezione e pilotaggio 25
B) Organi di lettura 26
C) Organi di inibizione 26
D) Forme d'onda di temporizzazione 26

3 L'UNITA' ARITMETICA 27

3.1. Generalità 27
3.1.1. Scopi ed operazioni consentite. 27
3.1.2. Codice ed organo di somma. 27
3.1.3. Organi principali dell'U.A. 30

3.2. Descrizione aritmetica delle operazioni con riferimento alle possibilità funzionali della macchina. 31
3.2.1. Somma di due numeri positivi 31
3.2.2. I segni 32
3.2.3. Sottrazione 33
3.2.4. Il contenuto dell'Accumulatore 34
3.2.5. Moltiplicazione additiva 36
3.2.6. Moltiplicazione sottrattiva 40
3.2.7. La fine delle operazioni 41
3.2.7.1. Somma e sottrazione 41
3.2.7.2. In Moltiplicazione. 42

3.3. Descrizione della esecuzione delle operazioni, con riferimento ai tempi ed agli organi interessati. 44
3.3.1. L'operazione di somma e sottrazione 44
3.3.2. L'operazione Moltiplicazione 46
3.3.3. L'operazione di trasferimento 51

3.4. Descrizione degli organi e del loro funzionamento con riferimento alle circostanze che lo determinano. 53
3.4.1. Ordine di Complementazione " a 9 " per la cifra proveniente da Memoria. 53
3.4.1.1. I segni
3.4.1.2. In moltiplicazione
3.4.2. Ordine " aggiungi unità " alla cifra proveniente da Memoria. 54
3.4.2.1. Riporto decimale di somma. 55
3.4.2.2. Complementazione a 10. 55
3.4.2.3. Temporizzazioni 56
3.4.3. Ordine di complementazione per la cifra letta in Accumulatore 57
3.4.4. Comando +1 sul canale di accumulatore 58
3.4.4.1. Generalità 58
3.4.4.2. Circostanza a) 58
3.4.4.3. Circostanza b) 59
3.4.5. Generatore di zeri in uscita dell'Accumulatore 59

4 GOVERNO 61

4.1. Premessa 61

4.2. Forme d'onda 62

4.2.1. Generalità 62
4.2.2. I Nastri 62
4.2.3. Gli impulsi di p.d.c. 62
4.2.4. I segnali di fase 63

4.3. Registri di Governo 64
4.3.1. Registri di Memoria 64
4.3.2. Registri di Accumulatore 65
4.3.3. Il registro di Lunghezza 66
4.3.4. Il registro di Funzione 67

5 DESCRIZIONE DELLE ISTRUZIONI IN FUNZIONE DEI TEMPI 68

5.1. Istruzione MA 68

5.1.1. Preparazione MA 69

5.1.2. Esecuzione MA 76

5.2. Istruzione MoA 80

5.2.1. Preparazione MoA 80

5.2.2. Esecuzione MoA 80

5.3. Istruzione AM 81

5.3.1. Preparazione AM 82

5.3.2. Esecuzione AM 82

5.4. Istruzione AoM 86

5.4.1. Preparazione AoM 86

5.4.2. Esecuzione AoM 86

5.5. Istruzione MT 87

5.5.1. Preparazione MT 87

5.5.2. Esecuzione MT 89

5.6. Istruzione MoT 91

5.6.1. Preparazione MoT 91

5.6.2. Esecuzione MoT 91

5.7. Istruzione TM 92

5.7.1. Preparazione TM 92

5.7.2. Esecuzione TM 92

5.8. Istruzione ToM 95

5.8.1. Preparazione ToM 95

5.8.2. Esecuzione ToM 95

5.9. Istruzione IT 96

5.9.1. Preparazione ed esecuzione IT 96

5.10. Istruzione +MA 99

5.10.1. Preparazione +MA 100

5.10.2. Esecuzione +MA 100
5.10.3. Procedura di fine 103

5.11. Istruzione -MA 114
5.11.1. Preparazione ed esecuzione -MA 114

5.12. Istruzione +MT 115
5.12.1. Preparazione +MT 115
5.12.2. Esecuzione +MT 115

5.13. Istruzione -MT 118
5.13.1. Preparazione -MT 118
5.13.2. Esecuzione -MT 118

5.14. Istruzione +IT 121
5.14.1. Preparazione ed esecuzione +IT 121

5.15. Istruzione -IT 124
5.15.1. Preparazione ed esecuzione -IT 124

5.16. Istruzione ITT 127
5.16.1. Preparazione ITT 127
5.16.2. Esecuzione ITT 127

5.17. Istruzione Y 130
5.17.1. Preparazione Y 130

5.18. Istruzione DA 133
5.18.1. Preparazione ed esecuzione DA 133

5.19. Istruzione FAM 135
5.19.1. Preparazione FAM 135
5.19.2. Esecuzione FAM 135

5.20. Istruzione CMA 139
5.20.1. Preparazione CMA 139
5.20.2. Esecuzione CMA 140

5.21. Istruzione CMT 141
5.21.1. Preparazione CMT 141

5.21.2. Esecuzione CMT 141

5.22. Istruzione CIT 142

5.22.1. Preparazione ed esecuzione CIT 142

5.23. Istruzione SC 143

5.23.1. Preparazione SC 143

5.23.2. Esecuzione SC 145

5.24. Istruzione SC 147

5.25. Istruzione SA 148

5.26. Istruzione SA 149

5.27. Istruzione S= 150

5.28. Istruzione S≠ 151

5.29. Istruzione STO 152

5.30. Istruzione SE1 153

5.31. Istruzione SE2 154

5.32. Istruzione SE3 155

5.33. Istruzione SE4 156

5.34. Istruzione STOP 157

5.35. Istruzione +X 158

5.35.1. Preparazione +X 159

5.35.2. Esecuzione +X 159

5.36. Istruzioni relative alle operazioni su nastro magnetico 171

5.37. Istruzione RNa 172

5.37.1. Preparazione RNa - fase 173

5.37.2. Esecuzione RNa - fase 173

5.37.3. fase 174

5.38. Istruzione RNi 176

5.38.1. Preparazione RNi 176
5.38.2. Esecuzione RNi 176

5.39. Istruzione LNa 177
5.39.1. Preparazione LNa - fase 178
5.39.2. Esecuzione LNa - fase 178
5.39.3. fase 179

5.40. Istruzione LNi 182
5.40.1. Preparazione LNi 182
5.40.2. Esecuzione LNi 182

5.41. Istruzione TN 183
5.41.1. Preparazione TN 183
5.41.2. fase 183

5.42. Istruzione DUB 185
5.42.1. Preparazione DUB 185
5.42.2. Esecuzione DUB 185

5.43. Istruzione NDNa 186
5.43.1. Preparazione NDNa 187
5.43.2. Esecuzione NDNa 187
5.43.3. Fase e 187

5.44. Istruzione NDNi 193
5.44.1. Preparazione NDNi
5.44.2. Esecuzione NDNi

1. LA MEMORIA

1.1. Generalità - Schema logico elementare

1.1.1. Definizione

La Memoria della Calcolatrice è l'organo che riceve informazioni e le trattiene per un tempo indefinito, sino ad ordine contrario, fornendole, ogni volta che sia richiesto, agli altri organi della macchina.

Le informazioni che la Memoria può conservare e fornire possono essere sia le istruzioni, che determinano i cicli di funzionamento della macchina, sia i dati su cui la macchina deve operare. Nella M. vengono pure immagazzinati risultati parziali e risultati definitivi delle operazioni svolte.

La caratteristica fondamentale che distingue la Memoria vera e propria da altri organi aventi funzioni analoghe nella Calcolatrice (nastro magnetico, tamburo magnetico, etc.) è la velocità a cui essa funziona; infatti il tempo di operazione su un bit in Memoria (e cioè: tempo di accesso ad un indirizzo, lettura, scrittura - o riscrittura) è dell'ordine di pochi microsecondi.

1.1.2. Capacità della Memoria

- La M. può immagazzinare 10.000 caratteri (di 5 bit ciascuno), in altrettante celle, ognuna delle quali è individuata mediante un numero d'ordine (indirizzo). I 10.000 indirizzi delle celle della M. sono compresi tra il numero zero e il 9999; ogni volta che un indirizzo compreso tra i precedenti è presentato dai registri W alla M., viene selezionata la cella corrispondente.
- Vi sono inoltre indirizzi fittizi, ai quali cioè non corrisponde la selezione di alcuna cella di M. Un indirizzo è fittizio ogni volta che uno almeno dei registri W presenta la configurazione non decimale 0000 oppure 1000 .
- Una cella di M. per volta è indirizzabile, ossia la M. opera le informazioni un carattere per volta (cioè a blocchi di 5 bit in parallelo, di cui 4 costituiscono l'informazione vera e propria, mentre il 5° ha funzione di controllo sul signifi-

cato dei precedenti quattro).

1.1.3. Ciclo fondamentale di funzionamento della Memoria

- Dalla precedente esposizione risulta che il ciclo fondamentale di funzionamento della Memoria consiste nell'estrarre e/o immagazzinare un carattere (di 5 bit) in una cella prescelta.
- L'indirizzo della cella prescelta è presentato in modo continuo dai registri W agli organi di selezione della Memoria.
- Un ciclo di operazione della Memoria ha inizio quando si dà il comando di pilotaggio della Memoria.
- Nel ciclo distinguiamo due fasi, succedentesi nel tempo, rispettivamente la lettura (o interrogazione), prima, della cella selezionata, e la scrittura, poi, nella cella stessa.
- Questo ciclo di lettura-scrittura si effettua sempre ed automaticamente, una volta dato il comando di pilotaggio, a partire da M 4,5 .

INTERROGAZIONE (o lettura)

Questa fase occupa la M. dal M 4,5 al M7 (Pilotaggio lettura).

- Il carattere contenuto all'indirizzo selezionato è reso disponibile al M6 da un comando sonda; ed - attraverso l'orgate Ou - staticizzato negli organi di uscita S della M.
- Se il comando di sonda non viene dato, il carattere contenuto in M. non appare all'ingresso degli staticizzatori, e quindi in uscita dalla M.

  In questo caso negli organi di uscita può essere staticizzato un carattere proveniente o dai registri W o dalla Console, i cui canali confluiscono nell'or Ou con quello proveniente da M. all'ingresso degli staticizzatori di uscita.
- Ovviamente i comandi di sonda, staticizzazione da W e staticizzazione da Console, si escludono a vicenda ed uno solo per ciclo può essere presente.
- L'informazione in uscita degli staticizzatori dura dal M6 al M5 del successivo p.d.c., allorchè gli staticizzatori sono cancellati.

SCRITTURA

Questa fase occupa la M. dal M7,5 al M10 (Pilotaggio scrittura).

- L'informazione da scrivere può provenire sia dall'esterno (tramite U.A.), sia dal canale di rigenerazione collegato con la uscita della M. stessa.
- Nel caso di scrittura attraverso il canale di rigenerazione, si immagazzina in M., il contenuto stesso della M. (riscrittura).
- Ovviamente la scrittura dall'esterno esclude il canale di rigenerazione e viceversa; ossia dei due comandi: scrittura dall'esterno, riscrittura, uno solo per ciclo può essere presente.
- La fig. 1.1. rappresenta lo schema logico elementare della M.; la fig. 1.2. rappresenta il diagramma temporale di un ciclo della M.

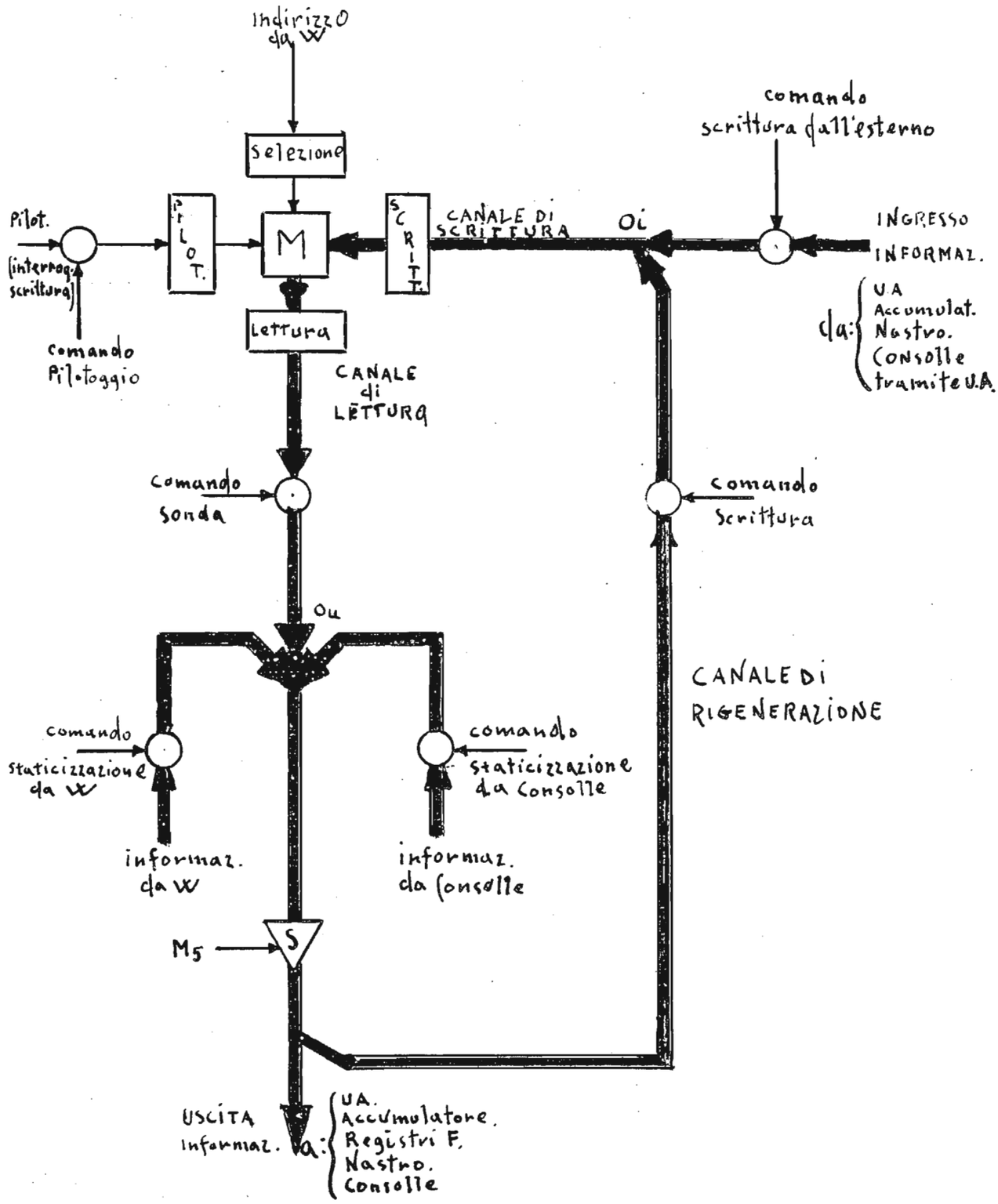


Fig. 1.1 SCHEMA LOGICO ELEMENTARE DELLA MEMORIA

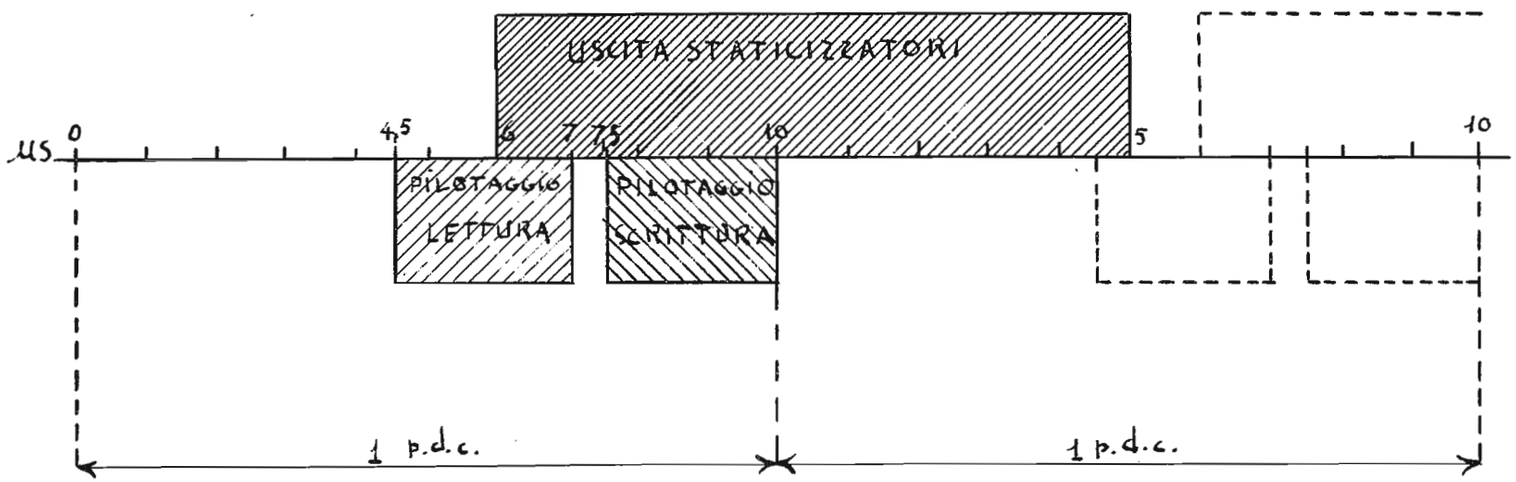
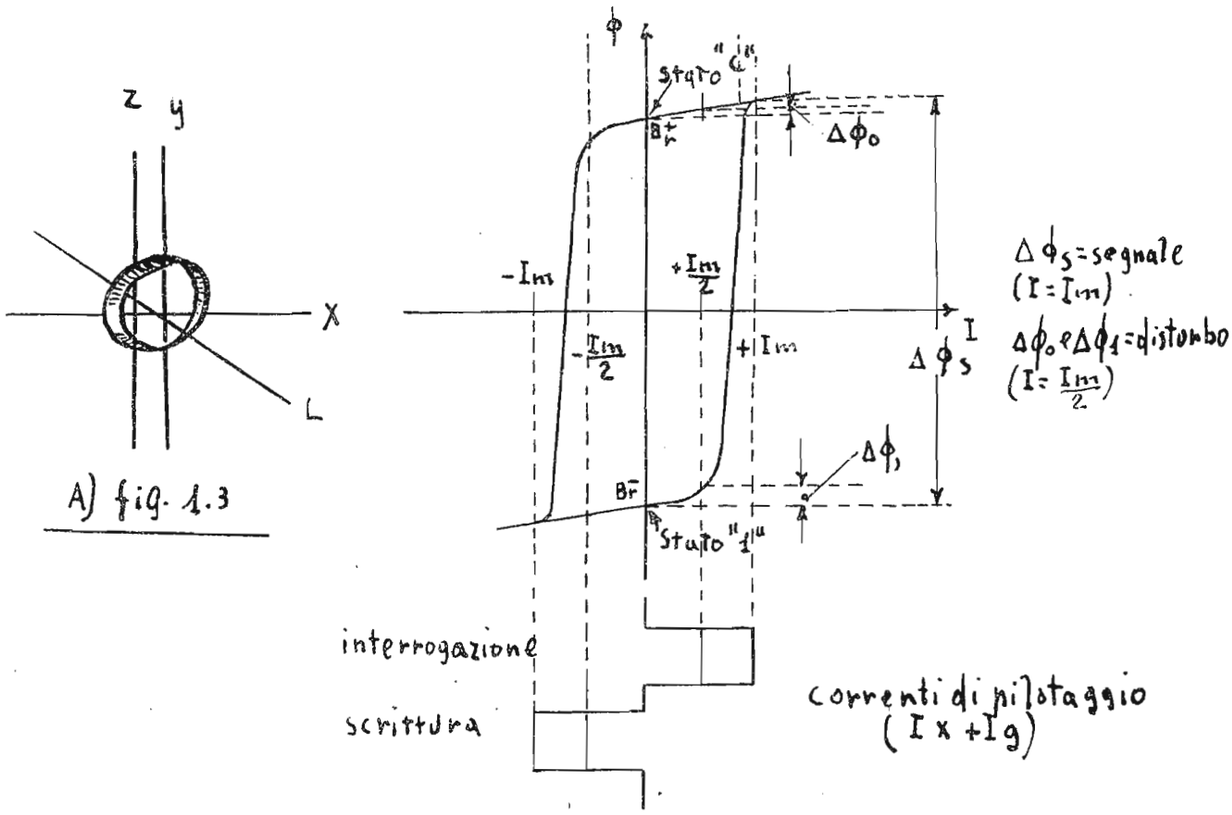


Fig. 1.2 DIAGRAMMA
TEMPORIZZAZIONI DELLA MEMORIA



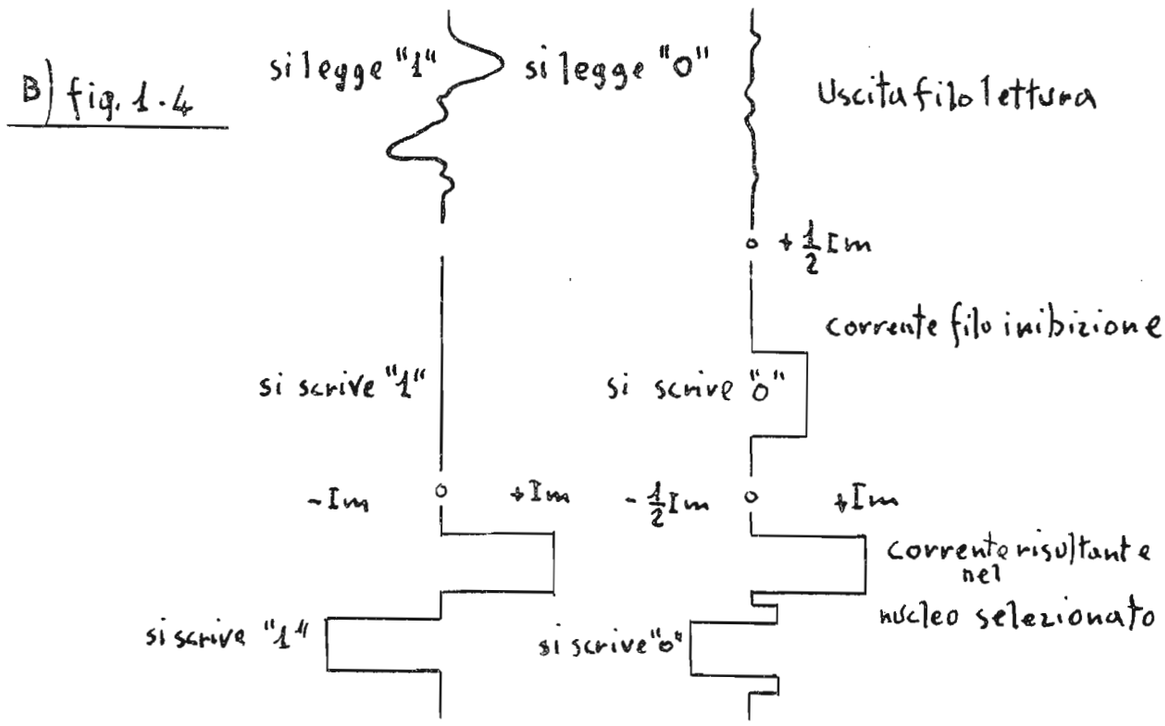


Fig. 1-3 e 1-4 – PRINCIPI FISICI DI FUNZIONAMENTO DELLA MEMORIA

1.2. Principi fisici di funzionamento della Memoria

1.2.1. Elemento fisico fondamentale di Memoria

L'elemento fisico fondamentale della M., ossia la cellla elementale capace di immagazzinare per un tempo indefinito una informazione binaria ("0", "1") e capace di commutare - a comando - tra l'uno e l'altro dei due stati in un tempo dell'ordine del microsecondo, è costituito da un nucleo toroidale di materiale magnetico avente un ciclo di isteresi rettangolare (figg. 1.3. e 1.4.).

La rettangolarità dei cicli di isteresi corrisponde infatti alla esistenza di due stati stabili di magnetizzazione del nucleo, corrispondenti ai due stati di saturazione positiva e negativa. Lo stato di rimanenza positiva B_r^+ rappresenterà allora per es. l'informazione "0", e lo stato di rimanenza negativa B_r^- rappresenterà quindi l'informazione "1".

Data la rettangolarità del ciclo di isteresi, è evidente che un nucleo rimarrà praticamente nello stato di saturazione in cui si trova finchè si applica un campo magnetizzante di grandezza inferiore al valore coercitivo Hc . Ciò permette un comodo assiemaggio delle celle costitutive della M.; infatti i nuclei possono essere disposti in matrici bidimensionali secondo due assi ortogonali xy (fig. 1.5.), in modo che entro ciascuna matrice ogni nucleo è univocamente individuato dalle sue coordinate xy . E' allora possibile selezionare un nucleo ed uno solo della matrice inviando su entrambe le linee xy che lo individuano, un impulso di corrente Ix rispettiv. Iy di valore tale che ciascuno di essi, preso singolarmente, non alteri lo stato di saturazione del nucleo (cioè $Ix = Iy < Im$, essendo Im il valore di corrente occorrente per portare in saturazione il nucleo), mentre la coincidenza delle due correnti Ix e Iy provochi la commutazione del nucleo da uno stato di saturazione all'altro (cioè $Ix + Iy > Im$).

Si è scelto per la nostra memoria un valore $Ix = Iy = \sim \frac{1}{2}\ Im$.

La "lettura" dell'informazione contenuta in un nucleo disposto in matrice si può effettuare mandando, sempre per coincidenza, un impulso di corrente Im , di polarità conosciuta e costante (impulso di lettura) e rilevando con opportuni mezzi se questo impulso ha causato oppure no una transizione del nucleo da uno stato di saturazione all'altro. Poichè al cambiamento di stato corrisponde una variazione di flusso nel nucleo, il rilevamento della eventuale transizione può essere fatto in base alla presenza o meno di un impulso di tensione $e = d\phi/dt$ indotto in

un altro avvolgimento concatenato col nucleo (filo di lettura).

Dalla fig. 1.4. si osserva che si ha l'impulso sul filo di lettura quando il nucleo si trova in stato "1"; se il nucleo si trova in stato "0" l'impulso di corrente Im di lettura non cambia praticamente lo stato di rimanenza magnetica del nucleo, e quindi non si ha praticamente impulso di tensione indotta sul filo di lettura.

L'impulso Im di lettura permette quindi di riconoscere lo stato in cui si trova il nucleo interrogato; è chiaro però che la lettura del nucleo è distruttiva dell'informazione in esso contenuta, nel caso che si legga un "1".

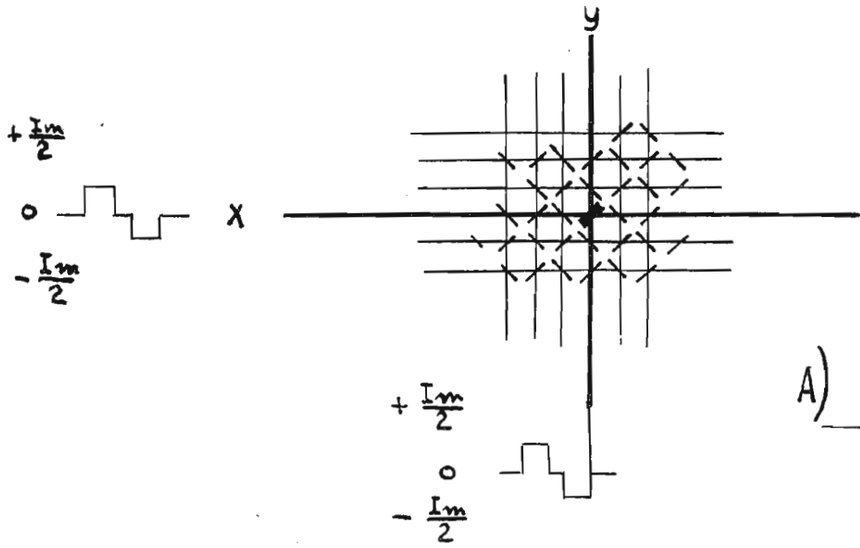
Per rigenerare l'informazione letta nel nucleo, a ciascun impulso di corrente di lettura si fa sempre seguire un impulso di corrente di uguale ampiezza ma di senso opposto (impulso di scrittura); nel caso si fosse letto "1" è chiaro che l'impulso di scrittura ripristina l'informazione nel nucleo. Però se l'informazione letta è uno "0", l'impulso di scrittura distruggerebbe l'informazione stessa. Questa difficoltà è superata infilando il nucleo con un altro avvolgimento z (filo di inibizione) e inviando su di esso, contemporaneamente all'impulso di scrittura, un impulso di corrente Iz di valore uguale a quello di Ix = Iy , ma di senso tale da creare un campo di direzione opposta. E' chiaro allora che l'effetto dell'impulso Iz è di inibire la transizione del nucleo dallo stato "0" allo stato "1" in corrispondenza dell'impulso di scrittura, in quanto il nucleo viene operato solo da una mezza corrente di scrittura.

In conclusione il processo di interrogazione non distruttiva di una cella di memoria consiste nell'inviare sempre gli impulsi Ix Iy di lettura e di scrittura; di raccogliere l'informazione sul filo di lettura; ed infine di inviare o meno l'impulso di inibizione in funzione della lettura fatta.

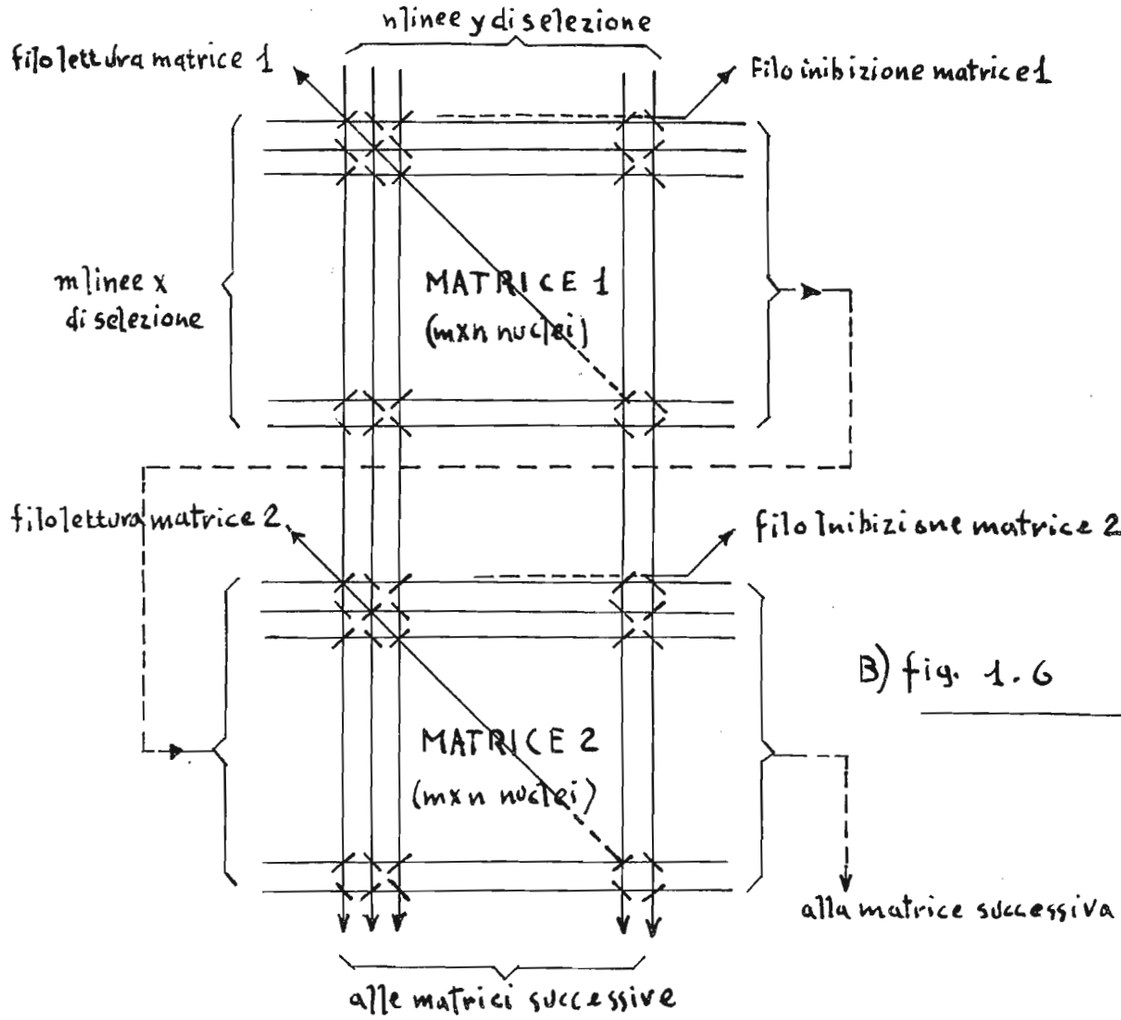
E' evidente anche che se l'impulso di inibizione viene comandato, anzichè dalla lettura della M., dall'esterno della M., si può immagazzinare nel nucleo a volontà "1" (= assenza di inibizione) o "0" (= presenza di inibizione).

Abbiamo dunque visto come ogni nucleo di materiale magnetico a ciclo di isteresi rettangolare possa essere usato come cella di memoria di informazioni binarie, organizzata in matrici bidimensionali.

Le matrici bidimensionali possono essere a loro volta collegate serializzando i fili x ed i fili y omonimi (fig. 1.6.).



A) fig. 1.5



B) fig. 1.6

Fig. 1.5 e 1.6 - PRINCIPIO DI MATRICE DI NUCLEI DI MEMORIA

Si ha in tal modo una organizzazione delle celle di memoria in un reticolo a 3 dimensioni, con possibilità di accesso in parallelo a tante celle quante sono le matrici.

In altri termini ogni nucleo è individuato, agli effetti della selezione, dai tre fili x y z , i primi due in comune con tutte le matrici; ed il terzo proprio di ciascuna matrice. L'avvolgimento di lettura poi è ovviamente proprio di ciascuna matrice.

1.2.2. Principio di anticoincidenza nella selezione della memoria.

Abbiamo visto (fig. 1.5.) come, disponendo i nuclei di memoria entro una matrice bidimensionale, la selezione di uno qualsiasi di essi possa effettuarsi inviando lungo due linee x e y due correnti uguali in valore ed in senso, in modo da avere la selezione per coincidenza del nucleo posto all'incrocio.

La selezione di un nucleo qualsiasi - e di uno solo per volta - entro la matrice è dunque legata alla selezione di due linee coordinate, una x ed una y .

Si può ora ottenere lo stesso risultato selezionando tre anzichè due linee coordinate, per es. due linee x ed una y . Ciò è possibile se le due linee x sono tra di loro in anticoincidenza, tali cioè che rispetto alla corrente Iy solo una delle correnti x abbia senso tale da dar luogo a coincidenza (sia in lettura che in scrittura).

Si ha pertanto la selezione di un nucleo e uno solo della matrice, pur attivando contemporaneamente tre linee coordinate di memoria.

In fig. 1.7. è mostrato uno degli schemi che realizzano il principio esposto. Se in una matrice coi nuclei disposti come nella precedente (fig. 1.5.), colleghiamo le linee x a coppie, come x_n e x_{n+2} , in serie tra loro, esse risultano automaticamente tra di loro in anticoincidenza rispetto a qualsiasi linea coordinata y . Infatti se la corrente in y ha il senso Iy , il nucleo selezionato è C ; se invece, tenendo fisso il senso delle correnti Ix , la corrente in y ha il senso di I'_y , il nucleo selezionato è C' .

E' chiaro da ciò che per selezionare tutti i nuclei di una matrice usando questo sistema, basterà, tenendo fisso il senso della corrente ad es. sulle linee x , invertire quello sulle linee y .

Il vantaggio di questo sistema risulta dal fatto che le linee riunite a coppie sono serializzate tra di loro, per cui basta un unico generatore di corrente per la selezione ed il pilotaggio di entrambe. In altre parole il numero di generatori di selezione e pi

lotaggio dell'asse x viene ridotto del 50% rispetto al sistema convenzionale illustrato in precedenza.

1.2.3. Principio di selezione mediante sottomatrici.

L'elemento di selezione e pilotaggio delle linee coordinate xy delle matrici di memoria è costituito (fig. 1.8.) da un trasformatore di impulsi con presa centrale sul primario; i due capi del primario sono collegati a due generatori di corrente Gk e G'k , mentre il secondario è collegato ad una linea y oppure ad una coppia di linee x , secondo il principio di anticoincidenza.

Il trasformatore a presa centrale permette di inviare su un unico filo collegato col secondario i due impulsi opposti di lettura e di scrittura. Infatti attivando prima il generatore Gk e poi G'k , o viceversa, si ottengono sull'unico secondario due impulsi di senso opposto tra loro, di cui il primo in ordine di tempo è quello di lettura.

E' evidente che per pilotare direttamente matrici di memoria di n x n coordinate, occorrono (col sistema ad anticoincidenza) $(n + \frac{n}{2})$ di questi generatori elementari, ossia $\frac{3}{2}$ n trasformatori e 3 n generatori di corrente G .

Si può **ulteriormente** ridurre l'onere del pilotaggio introducendo il principio di sottomatrice: cioè ordinare i generatori elementari in matrici, una per il pilotaggio delle linee x ed una per le linee y delle matrici di nuclei.

In ognuna di queste sottomatrici i trasformatori sono individuati da due linee coordinate, passanti rispettivamente per il centro e per i due estremi del primario (v. fig. 1.9.); è evidente allora che un trasformatore è selezionato se entrambe le sue due linee coordinate sono selezionate contemporaneamente. Ogni altro trasformatore appartenente ad una linea della sottomatrice attivata, ma per il quale non si verifichi la coincidenza, non è selezionato; infatti i generatori come Gh e (Gk , G'k) costituiscono due interruttori in serie al primario di ciascun trasformatore.

Il vantaggio di questo sistema è che, fermo restando il numero di trasformatori occorrenti, il numero dei generatori G si riduce all'incirca nel rapporto da n a $\sqrt{n}$.

Notare (fig. 1.9.) l'inserzione ai capi del primario di ogni trasformatore di un elemento separatore (diodo D), per impedire selezioni spurie dei trasformatori.

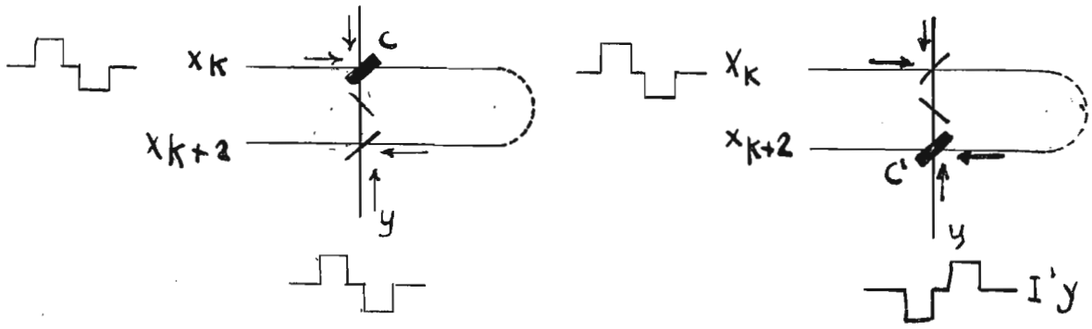
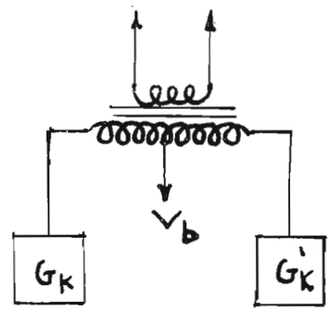


Fig. 1-7
PRINCIPIO DI ANTICOINCIDENZA

Fig. 1-8- GENERATORE IMPULSI BIDIREZIONALI



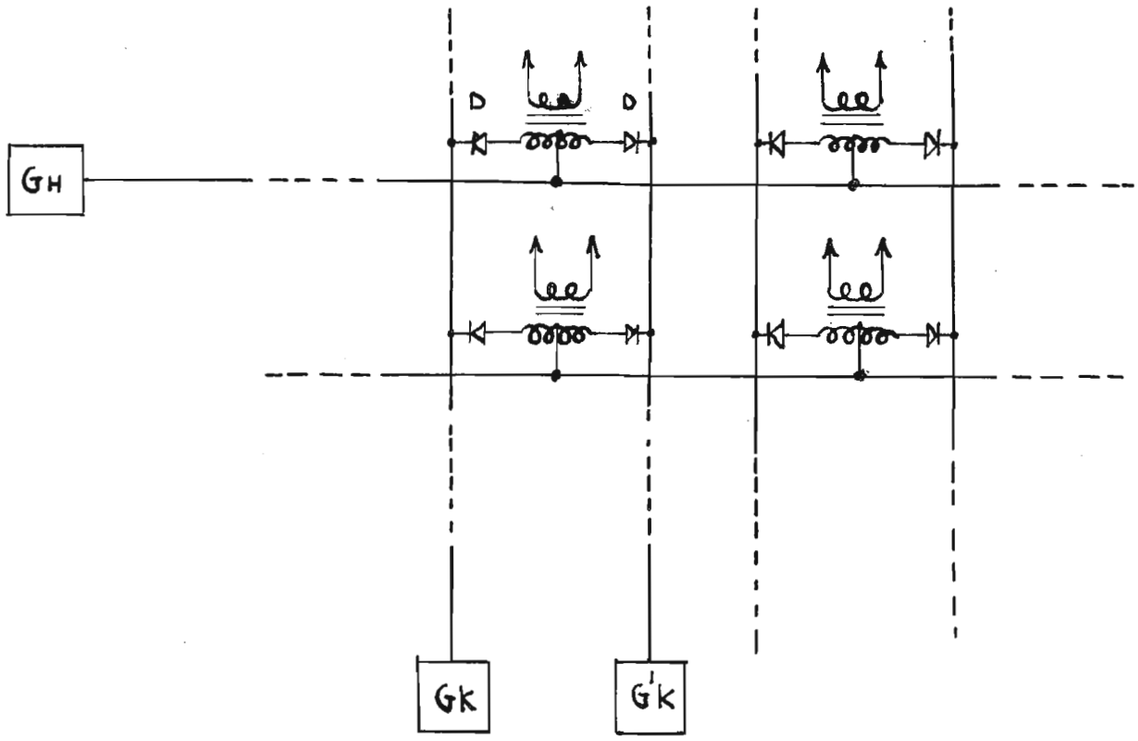


Fig. 1.9 _ PRINCIPIO DI SOTTOMATRICE DI PALOTAGGIO

1.2.4. Principio di lettura in una matrice mediante discriminazione temporale.

Nel par. 1.2.1. si è illustrato il principio su cui si basa la lettura dell'informazione contenuta in un nucleo; si è visto come la lettura di un "1" corrisponde ad un impulso di tensione indotto in un avvolgimento di senso (filo di lettura), mentre la lettura di uno "0" corrisponde praticamente per un nucleo singolo a mancanza di impulso sul filo di lettura. In altri termini il riconoscimento dell'informazione contenuta in un nucleo potrebbe essere fatto mediante semplice discriminazione di ampiezza sui segnali ottenuti sul filo di lettura.

Ora questa semplice discriminazione valida per un nucleo singolo, non è più sufficiente quando il nucleo si trovi assiemato ad altri entro una matrice bidimensionale (come visto in precedenza), in cui il filo di lettura è unico e concatenato con tutti i nuclei della matrice stessa.

Infatti se per un nucleo, preso singolarmente, il rapporto tra la tensione corrispondente ad un "1" e quella di uno "0", ossia brevemente il rapporto segnale/disturbo è assai grande, questo stesso rapporto quando il nucleo interrogato è disposto in una matrice, si riduce fino anche a capovolgersi, per effetto dei disturbi indotti sul filo di lettura da tutti gli altri nuclei della matrice appartenenti alle linee selezionate, ma per i quali non si verifica la coincidenza.

Infatti un nucleo pilotato con una sola delle correnti di selezione, cioè $\frac{1}{2}$ Im, genera, a causa della non perfetta rettangolarità del ciclo di isteresi (v. fig. 1.4.), una variazione di flusso, piccola rispetto a quella dovuta alla transizione da uno stato di saturazione all'altro, ma non nulla.

Poichè il disturbo complessivo raccolto sul filo di lettura è la somma algebrica dei singoli disturbi, l'avvolgimento di lettura è disposto secondo particolari percorsi in modo da raccogliere le tensioni indotte da metà nuclei della matrice con polarità opposta a quella indotta dall'altra metà.

Malgrado questo accorgimento, però, la somma algebrica dei disturbi non ha avuto valore nullo; la non perfetta elisione dei disturbi è dovuta - a parte le inevitabili differenze fisiche tra nucleo e nucleo - essenzialmente alla asimmetria del ciclo di isteresi rispetto all'asse I , per cui una stessa corrente $+\frac{1}{2}$ Im causa una variazione di flusso diversa a seconda che il nucleo si trovi nello stato "1" oppure nello stato "0" ($\Delta \phi_1 \neq \Delta \phi_0$; vedi fig. 1.4.).

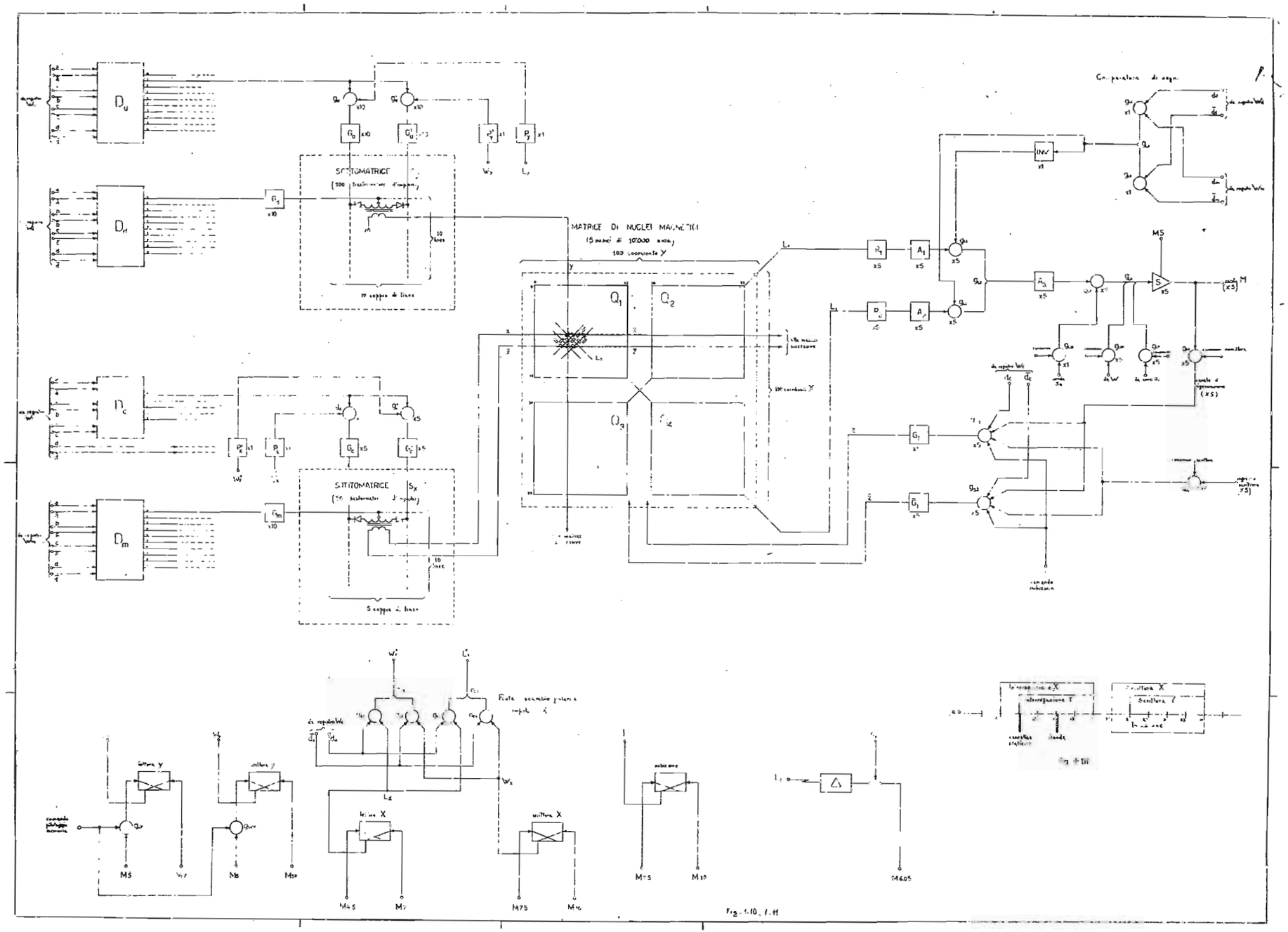
Vi è pertanto in ogni matrice un disturbo complessivo dato dalla sommatoria delle differenze di disturbi dei due tipi. Il valore di picco di tale disturbo dipende quindi volta per volta dal particolare contenuto di informazione dei nuclei della matrice; e può essere, per matrici con numero molto grande di nuclei, dello stesso ordine di grandezza del segnale prodotto dal nucleo selezionato.

La discriminazione del segnale dal disturbo è però ancora possibile se la misura di ampiezza viene associata ad un altro parametro, come il tempo.

Il sistema da noi adottato è basato sul fatto che,mentre il segnale vero e proprio raggiunge il valore di picco in un tempo dipendente praticamente dal tempo di orientamento dei domini magnetici elementari del materiale (tempo di commutazione proprio del nucleo), il quale è funzione essenzialmente del valore della corrente di pilotaggio, il disturbo invece ha un tempo di salita che è circa uguale a quello dell'impulso di corrente di pilotaggio (infatti per $I < Im$, si ha praticamente un accoppiamento a trasformatore unitario con $k = \sim 1$, tra filo di pilotaggio e filo di lettura).

Pertanto essendo il tempo di peaking dei nuclei da noi usati dell'ordine di 1 μs , se si usano impulsi di corrente di pilotaggio con fronti sufficientemente ripidi, si ha il margine per una discriminazione temporale di ampiezza tra segnale e disturbo in una matrice.

In conclusione la lettura in una matrice di memoria viene effettuata "sondando" ad un istante opportuno e prefissato la tensione raccolta sul filo di lettura.


MATRICE DI NUCLEI MAGNETICI
SOTTOMATRICE
SOTTOMATRICE
Comparatore di segno
Rete scambio potenza
Fig. 4.10, 4.11

1.3. Schema funzionale a blocchi della Memoria

1.3.1. Introduzione

Le caratteristiche funzionali della M. possono essere riassunte nei seguenti punti:

1) Capacità di selezionare qualsiasi cella di M. in base allo indirizzo fornito dai registri W ;

2) capacità di leggere il contenuto della cella, e di presentarlo agli altri organi della macchina in forma conveniente per essere utilizzato;

3) capacità di immagazzinare in una sua cella qualsiasi una qualsiasi informazione (proveniente da altri organi della macchina o della M. stessa), è di trattenerla per un tempo indefinito fino ad ordine contrario.

In base a queste caratteristiche funzionali, gli organi interni costituenti la M. possono raggrupparsi nel seguente modo:

A) organi di decodificazione, di selezione e di pilotaggio delle linee coordinate di Memoria;

B) organi di riconoscimento e formazione dei segnali di lettura;

C) organi di inibizione, con comando esterno o interno (rigenerazione).

I tre organi sopra elencati sono poi organizzati da un :

D) organo generatore e distributore di segnali di comando e temporizzazione.

La fig. 1.10. rappresenta lo schema a blocchi della M.
La fig. 1.11. rappresenta i segnali di comando e temporizzazione dentro un periodo di cifra.

Partendo dalla sinistra della fig. 1.10. e procedendo verso destra noi troviamo successivamente gli organi della M. nell'ordine A), B), C), D).

Al centro tra gli organi A) ed i rimanenti, è collocato l'organo di memoria vero e proprio, cioè le matrici di nuclei magnetici; il disegno schematizza una delle matrici, mostrandone i collegamenti essenziali per la comprensione del funzionamento della Memoria.

Nella descrizione dello schema a blocchi della M. noi inizieremo illustrando la memoria vera e propria, cioè l'insieme dei nuclei magnetici; seguirà poi la descrizione ed il funzionamento degli organi da A) a C), secondo la temporizzazione stabilita dall'organo D).

1.3.2. Matrici di nuclei magnetici

Nel cap. 1.2. si è stabilito che i nuclei costituenti la M. debbono essere disposti in un reticolo a tre dimensioni, costituito da matrici bidimensionali collegate serialmente tra loro.
La M. consta di 5 di tali matrici. Ogni matrice è composta di 10.000 nuclei, secondo un quadro di 100 x 100 linee coordinate x e y .

Le linee x e y sono, come già detto, serializzate tra le matrici, per cui in definitiva il pilotaggio di lettura e scrittura dell'intera M. è ricondotto al pilotaggio di 100 linee x e 100 linee y .

Inoltre, in base al principio di anticoincidenza, il pilotaggio lungo l'asse x è ricondotto al pilotaggio di 50 coppie di linee come x_n e x_{n+2} , tra loro serializzate.

In conseguenza dell'applicazione dell'anticoincidenza, il filo di inibizione z non può essere unico per tutti i nuclei di una matrice; infatti si può constatare facilmente che comunque lo si disponga un unico filo z che infilasse tutti i nuclei della matrice potrebbe inibire in ogni coppia x_n , x_{n+2} . La linea selezionata però provocherebbe transizioni spurie nei nuclei dell'altra linea (attivata ma non selezionata) della coppia, essendo in questa la corrente Iz necessariamente in coincidenza con l'impulso di scrittura.

Pertanto il filo di inibizione è diviso in due parti, si hanno cioè due avvolgimenti separati, ognuno dei quali attraverso metà dei nuclei della matrice, in modo tale che in ogni coppia di linee come (x_n , x_{n+2}) operate contemporaneamente, una linea appartenga ad un avvolgimento di inibizione "z, e l'altra ad un altro avvolgimento di inibizione $\bar{z}$". E' chiaro allora che per evitare l'inconveniente in questione, basterà mandare l'impulso di corrente di inib. su uno solo dei due avvolg. z $\bar{z}$, e precisamente su quello cui appartiene la linea x effettivam. selez. della coppia.

Anche l'avvolgimento di lettura della matrice è sdoppiato in due avvolgimenti separati L_1 e L_2 , ognuno dei quali comprende una metà dei nuclei della matrice, secondo lo schema rappresentato

in fig. 1.10. Rispetto ai fili di lettura la matrice è divisa in quattro quadranti Q_1 ... Q_4 , di cui Q_1 è collegato con Q_4 , mentre Q_2 è collegato con Q_3 . Lo scopo di questo sistema è di ottenere un netto miglioramento del rapporto segnale/disturbo , che si rende necessario dato il numero di nuclei nella matrice.

Infatti usando un unico avvolgimento di lettura, i nuclei disturbati (cioè attivati con corrente metà) sono praticamente 2 x 100 sull'asse x + 100 sull'asse y ; ossia in totale sul filo di lettura si raccoglierebbero 300 disturbi; ed è questo un valore eccessivo per una sicura discriminazione del segnale dal disturbo.

Se ora dividiamo il filo di lettura secondo lo schema visto, noi avremo sempre su ciascun filo di lettura la metà dei disturbi complessivi nella matrice (2 x 50 nuclei sull'asse x + 50 sull'asse y , ossia in totale 150).

Se allora noi consideriamo di volta in volta soltanto quell'avvolgimento di lettura cui appartiene il nucleo selezionato è chiaro che aumentiamo del 100% il rapporto segnale/disturbo. In altri termini basterà di volta in volta leggere il filo di lettura entro cui avviene la coincidenza delle correnti xy , e soltanto quello.
(Nota - Un ulteriore accorgimento per migliorare il rapporto segnale-disturbo di una matrice verrà illustrato nel paragrafo seguente 1.3.3.).

In conclusione possiamo stabilire il seguente quadro dell'organo "nuclei di memoria":

nuclei totali :	50.000
matrici :	5 , ciascuna di 10.000 nuclei (in quadro 100 x 100)
fili di pilotaggio:	100 x (= 50 coppie) e 100 y , comuni a tutte le matrici.
fili di inibizione:	2 per ciascuna matrice.
fili di lettura :	2 per ciascuna matrice.

1.3.3. Selezione e pilotaggio della Memoria

In fig. 1.10., partendo da sinistra, noi troviamo anzitutto quattro gruppi di quattro canali binari, costituenti le uscite rispettivamente dei registri di Memoria Wu Wd Wc e Wm .
Questi registri presentano in modo continuo agli organi di selezione della M. l'indirizzo della cella (5 nuclei in parallelo) da selezionare; Wu presenta la cifra delle unità dell'indirizzo, Wd la cifra decine, Wc le centinaia e Wm le migliaia.

Da ogni registro escono quattro canali binari $a\bar{a}$, $b\bar{b}$, $c\bar{c}$, $d\bar{d}$, che costituiscono una cifra decimale, secondo il codice adottato nella calcolatrice.

L'indirizzo binario non è però direttamente utilizzabile per la selezione della M. Infatti se teniamo presente il principio di selezione illustrato nel paragrafo 1.2.3. noi abbiamo che selezionare l'indirizzo (decimale) m c d u , significa selezionare la linea orizzontale d , e la linea verticale u della sottomatrice Sy , e la linea orizzontale m e la linea verticale c della sottomatrice Sx .

Cioè ad ogni configurazione delle uscite binarie di un registro, deve corrispondere un filo ed uno solo selezionato nella sottomatrice corrispondente.

E' pertanto necessario inserire tra ogni registro W e gli organi di selezione della relativa sottomatrice, un decodificatore da binario a decimale, di tipo convenzionale, quale sarà descritto nella parte III^ .

Sullo schema a blocchi troviamo pertanto i quattro decodificatori Du Dd Dc Dm corrispondenti rispettivamente ai registri di unità, decine, centinaia e migliaia.

I decodificatori Du Dd e Dm sono tra loro uguali e decodificano quattro canali binari in una su 10 linee ciascuno; il registro Dc decodifica solo tre canali binari del registro relativo, in una su 5 linee; il quarto canale bianrio d , $\bar{d}$ del registro Wc viene direttamente utilizzato nella selezione della M.; infatti le linee d , $\bar{d}$ di Wc costituiscono due dei quattro ingressi della rete di scambio della polarità degli impulsi di lettura-scrittura, secondo il principio di anticoincidenza. La rete di scambio è costituita dai quattro and gates g_x1 , g_x2 , g_x3 , g_x4 e dai due or gates g_x5 e g_x6 .

Allorchè il comando di interrogazione viene applicato al terminale Lx , esso passerà sul canale di uscita dell'or g_x5 oppure dell'or g_x6 a seconda la configurazione del canale d , $\bar{d}$ del registro Wc . Analogamente il comando di scrittura applicato al terminale Wx passerà sul canale di uscita dell'or g_x6 oppure dell'or g_x5 .

Proseguendo nello schema verso destra, troviamo le due sottomatrici di trasformatori di pilotaggio coi relativi generatori.

La sottomatrice Sy consta di 100 trasformatori di impulsi a presa centrale (coi relativi diodi separatori), disposti in un quadro di 10 x 10 . La sottomatrice Sx consta di 50 trasformatori, disposti secondo un rettangolo di 10 x 5 .

Ogni trasformatore è collegato ai generatori G della sottomatri-

ce cui appartiene; precisamente i centri del primario sono collegati ai generatori come Gd e Gm, rispettivamente per la sottomatrice y e x ; mentre i due estremi del primario sono collegati ad una coppia di generatori come Gu e G'u per la sottomatrice Sy e Gc e G'c per la Sx . (gli indici u, d, c, m, indicano la corrispondenza con i rispettivi decodificatori e registri).

E' chiaro, da quanto illustrato nel cap. 1.2., che vi sono complessivamente: 10 coppie di generatori Gu e G'u , 5 coppie di G G'c, 10 generatori Gd e 10 Gm .

Ogni trasformatore della sottomatrice Sy ha il secondario collegato ad una linea y delle Metrici di M.; invece nella sottomatrice Sx ogni secondario è collegato a due linee x , tra loro in serie.

E' importante notare subito il modo differente con cui sono selezionati i generatori tipo Gu G'u e Gc G'c da quello che avviene per i generatori come Gd e Gm .

Infatti mentre questi ultimi vengono direttamente selezionati dal rispettivo decodificatore, per i generatori del tipo Gu G'u e Gc G'c la selezione è legata alla coincidenza negli and gates siglati g_u g'_u e g_c g'_c , tra il segnale proveniente dal decodificatore e un segnale di consenso proveniente rispettivamente dai blocchi piloti Py P'y e Px P'x . Questo segnale di consenso costituisce la temporizzazione del pilotaggio della M.

Gli and gates come g_u g'_u e g_c g'_c sono tanti quanti sono i rispettivi generatori, cioè 10 g_u , 10 g'_u , 5 g_c e 5 g'_c . I piloti P sono invece singoli, cioè in totale 4 per tutta la Memoria.

Possiamo ora descrivere il funzionamento dell'insieme dei blocchi costituenti la selezione ed il pilotaggio della M.

Sia presente nei registri W l'indirizzo da selezionare in M. Ciò significa che per ogni decodificatore D si ha una sola linea di uscita selezionata; risultano quindi selezionati i relativi generatori G . Però mentre i generatori, come Gd e Gm selezionati non sono attivati finchè non arrivano sui relativi and gates gli impulsi di temporizzazione.

In altre parole la selezione dei generatori di pilotaggio è effettuata con continuità, in base all'indirizzo presente nei W ; però il pilotaggio non si effettua fino a che non viene dato l'ordine di lettura e di scrittura (tramite i gates g_u g'_u e g_c g'_c).
Entro un periodo di cifra si è già visto (cap.1.1.) che la lettura occupa l'intervallo di tempo tra il μs4,5 ed il 7; mentre

la scrittura va dal us 7,5 al 10 .

In effetti il comando di lettura dura da 4,5 a 7 per la sottomatrice Sx , mentre per la sottomatrice Sy la lettura dura dal µs 5 al 7 .

Lo scopo di questo sfalsamento dell'inizio della lettura è di ottenere un miglioramento del rapporto segnale/disturbo delle matrici di M . Infatti secondo quanto enunciato nel par.1.2.4., i disturbi corrispondenti a mezza corrente di pilotaggio salgono all'incirca con la legge dell'impulso di corrente stesso; perciò se, noi mandiamo le due mezze correnti Iy e Ix in tempi successivi, anzichè contemporaneamente, abbiamo che i disturbi corrispondenti alla mezza corrente inviata per prima sono già in fase di discesa quando incominciano a salire i disturbi corrispondenti alla mezza corrente successiva. Se pertanto gli inizi delle due mezze correnti sono sufficientemente distanziati, si può considerare che all'istante in cui il segnale raggiunge il valore di picco (istante di sonda), il contributo di disturbi delle linee pilotate per prime sia nullo.

Noi abbiamo scelto di pilotare per prima la sottomatrice Sx , poichè si ottiene in tal modo il massimo guadagno di rapporto segnale/disturbo, essendo in ogni matrice pilotate contemporaneamente due linee x .
All'istante di sonda possiamo quindi considerare che il disturbo è essenzialmente quello dovuto alla linea y pilotata. In definitiva con questo accorgimento si migliora il rapporto segnale/disturbo all'istante di sonda nella misura di circa 2/3 .

Riprendendo la descrizione del funzionamento del pilotaggio, abbiamo che quando questo debba essere effettuato, all'istante 4,5 viene presentato al terminale Lx il segnale di comando di lettura delle linee x . Questo comando attraverso la rete di scambio già descritta, andrà a selezionare il generatore pilota Px oppure P'x a seconda la configurazione del canale d , $\bar{d}$ del registro Wc , ossia a seconda l'indirizzo da selezionare. Il generatore P attivato seleziona a sua volta allora, tramite i gates g_c g'_c , o il generatore G_c o G'_c . Si ha così al secondario del trasformatore appartenente alla riga ed alla colonna selezionate della sottomatrice, l'impulso di pilotaggio di lettura Ix .

All'istante 5 del periodo di cifra, viene presentato al terminale Ly il segnale di comando di lettura delle linee y .

Tramite il pilota Py viene allora selezionato il generatore G_u , e si ha al secondario del trasformatore scelto, l'impulso di pilotaggio di lettura Iy .

All'istante 7 il segnale di comando di lettura sia per Sx che per Sy ha termine; terminano di conseguenza gli impulsi di corrente di lettura sulle linee x e y .

La scrittura occupa come si è visto l'intervallo 7,5-10 del periodo di cifra. Anche in scrittura però gli impulsi sull'asse x sono più lunghi che sull'asse y , nella stessa misura che in lettura. La necessita di fare gli impulsi di scrittura della stessa durata di quelli di lettura è quella di non creare una componente continua di corrente nei trasformatori delle sottomatrici.

Il funzionamento del pilotaggio di scrittura è identico a quello di lettura. Al µs 7,5 al terminale Wx viene presentato il segnale di comando scrittura sulle linee x , il quale passa, attraverso la rete logica di scambio, a selezionare il pilota e conseguentemente il generatore che non erano stati attivati in lettura.

Al µs 8 al terminale Wy viene presentato il segnale di comando scrittura sulle linee y, che tramite il pilota P'y, attiva il generatore G'd .

Al µs 10 i segnali di comando in Wx e Wy hanno termine.
In definitiva si hanno sulle linee selezionate di M. dei segnali di scrittura aventi lo stesso valore e durata, ma senso opposto, a quelli inviati in lettura sulle stesse linee.

Abbiamo così esaurita la descrizione a blocchi del funzionamento degli organi di decodificazione, selezione e pilotaggio della Memoria.

1.3.4. Lettura della Memoria

Alla destra della matrice di Memoria schematizzata in fig.1.10., noi troviamo gli organi che servono al riconoscimento dei segnali raccolti sul filo di lettura della matrice, ed alla trasformazione dei segnali stessi nella forma standard, utilizzabile dagli altri organi della calcolatrice. Facciamo anzitutto notare che gli organi di lettura sono tanti quante sono le matrici di M., in quanto ogni matrice fornisce uno dei 5 bit che vengono letti in parallelo e costituiscono un carattere.

Si è già visto (par. 1.3.2.) come in ogni matrice vi siano due fili indipendenti di lettura, L_1 e L_2 , corrispondenti rispettivamente ai quadranti Q_2 Q_3 e Q_1 Q_4 della matrice (v.fig.1.10.), e come solo l'uno o l'altro di questi avvolgimenti debba essere volta per volta considerato in funzione dell'indirizzo selezionato.

Procedendo pertanto dalla matrice di Memoria verso destra noi troviamo due catene identiche costituite dai blocchi R_1 A_1 e R_2 A_2 , corrispondenti rispettivamente agli avvolgimenti L_1 e L_2 .

Le scatole R_1 e R_2 sono dei rettificatori a onda intera, la cui presenza è dovuta al fatto che i segnali sui fili di lettura possono presentarsi con polarità opposta.

Le scatole A_1 e A_2 sono amplificatori a larga banda; infatti i segnali all'uscita di un filo di lettura hanno un'ampiezza dell'ordine del centinaio di m V , e devono quindi essere amplificati opportunamente perchè si possa procedere alle successive manipolazioni.

L'uscita degli amplificatori A_1 e A_2 entra rispettivamente negli and gates gL1 e gL2 la cui uscita confluisce in or nel gate gL3 .

I gates gL1 e gL2 operano la scelta del filo di lettura, in quanto solo uno di essi lascia passare il segnale.
L'apertura o chiusura di ciascuno dei gates gL1 e gL2 è comandata dalla rete costituita dai gates gL4 gL5 gL6 e dallo invertitore Inv. I tre gates gL4 gL5 gL6 costituiscono un comparatore di segno tra il canale d , $\bar{d}$ del registro Wd e quello di Wm , nel senso che all'uscita della rete si ha il segnale di consenso solo quando entrambi gli ingressi dei gates gL4 e gL5 hanno la stessa polarità.

Ora in base al codice binario a simmetria quinaria della calcolatrice i canali d , $\bar{d}$ di Wd e Wm sono omopolari quando, e solo quando l'indirizzo corrispondente si trova nei quadranti Q_1 e Q_2 . In tal caso pertanto all'uscita dell'or gate gL6 si ha segnale di consenso che apre direttamente il gates gL2 ; nel gates gL1 arriva invece il segnale negato, data l'interposizione dell'invertitore Inv.

E' evidente che per ogni combinazione non omopolare dei canali d , $\bar{d}$ dei registri Wd e Wm, l'indirizzo selezionato si trova nei quadranti Q_2 e Q_3 , ed in tal caso risulta aperto il gates gL_1 .

All'uscita dell'or gate gL3 il segnale, dopo ulteriore amplificazione nella scatola A_3 , viene portato ad uno dei due ingressi dell'and gate gL7 . In questo gate avviene la discriminazione temporale del segnale dal disturbo, secondo i concetti esposti nel cap. 1.2.4.). Infatti all'altro ingresso del gate viene applicato ad un istante prefissato e costante un impulso di sonda, di breve durata (circa 0,3 μs). Questo impulso di sonda (che nella M. coincide col mastro M6) passerà o meno in uscita del gate gL7 a seconda il livello presente sull'ingresso da lettura del gate stesso.

Abbiamo pertanto che se il segnale letto è uno "0", il gate gL7 risulta chiuso per l'impulso sonda, che quindi non passa; se invece il segnale letto è un "1" il gate gL7 risulta aperto per l'impulso sonda, che passa attraverso il gate e, tramite l'or gate gL8, va a caricare lo staticizzatore temporaneo S . Questo staticizzatore è stato preventivamente cancellato, tramite l'altro suo ingresso, dal mastro M5 .

Pertanto, nel caso si legga "0" l'uscita dello staticizzatore non cambia; nel caso invece si legga "1", in corrispondenza del mastro M6 all'uscita dello staticizzatore compare il segnale di "1", nella forma e livello standard .

Va notato che, come accennato nel cap. 1.1., lo staticizzatore S può servire ad immagazzinare, anzichè la lettura da M., l'informazione impostata da Console o proveniente dai registri W . In tal caso il gate gL11 viene chiuso, per cui l'impulso sonda (M6) non passa; viene invece aperto il gate gL9 o gL10.

L'uscita dello staticizzatore S costituisce pertanto a tutti gli effetti logici e pratici l'uscita da Memoria.

1.3.5. Inibizione e scrittura della M.

Abbiamo descritto nel cap. 1.2. i principi su cui si fonda la scrittura nei nuclei di M., e cioè essenzialmente il concetto di inibizione dell'impulso di pilotaggio di scrittura. Nel paragrafo 1.3.2. abbiamo poi visto che in ogni matrice vi sono due avvolgimenti di inibizione distinti, concatenati ciascuno con una metà dei nuclei della matrice, dei quali uno solo per volta può essere percorso dall'impulso di inibizione, la scelta determinata dall'indirizzo di M. selezionato.

Nello schema di fig. 1.10. noi troviamo i due avvolgimenti di inibizione rappresentati dai due fili z e $\bar{z}$, coi relativi generatori G_I e $\bar{G}_I$.
La selezione dei generatori G_I e $\bar{G}_I$ avviene tramite i gates g_{s1} e g_{s2} , secondo lo schema seguente.

Ognuno degli and gates g_{s1} e g_{s2} ha quattro ingressi, e cioè: ingresso scrittura dall'esterno; ingresso scrittura da M. (riscrittura); ingresso comando inibizione (temporizzazione); ingresso da registro Wc (scelta del generatore in funzione dell'indirizzo).

La scrittura dall'esterno esclude la riscrittura, e viceversa;

provvedono a ciò i gates g_{s3} e g_{s4} , dei quali uno solo per volta può essere aperto.

L'ingresso dal registro Wc è costituito dal canale d , $\bar{d}$, lo stesso utilizzato nella rete di scambio degli impulsi di pilotaggio x (v. par. 1.3.3.).

Il comando di inibizione arriva al µs7,5 , in parallelo sui due gates g_{s1} , e g_{s2} ; uno solo però sarà in grado di lasciarlo passare, e cioè quello selezionato in base al canale d , $\bar{d}$ di Wc . Il passaggio del comando di inibizione al generatore relativo, è allora condizionato soltanto dall'informazione presente sul canale di scrittura, o di riscrittura. Precisamente, come sappiamo, l'impulso di comando passa ad attivare il generatore nel caso si debba scrivere "0", mentre nel caso di scrittura di "1" non si ha pilotaggio dell'inibizione.
Il comando pilotaggio inibizione ha termine al µs 10 di ciascun periodo di cifra.

1.3.6. Forme d'onda di temporizzazione

Nella fig. 1.11. sono rappresentate in modo schematico le forme d'onda di temporizzazione della M., il cui significato è stato illustrato singolarmente nella descrizione dello schema a blocchi della M.

Nella fig. 1.10., in basso, è disegnato l'organo generatore delle forme d'onda suddette. Si tratta sostanzialmente di flip flop triggerati da impulsi presi dalla linea di ritardo della calcolatrice; per i flip flop di lettura e scrittura delle linee x l'impulso di set è dotato di consenso. (comando pilotaggio M.).

L'identificazione dei generatori di forme d'onda di fig.1.10. è immediata, se confrontata col diagramma di fig.1.11.

Ci rimane solo da spiegare la funzione della scatola contrassegnata col simbolo Δ .
All'ingresso della scatola è applicato un impulso ottenuto differenziando la forma d'onda di comando interrogazione Ly . Nella scatola l'impulso subisce un ritardo Δt, il cui valore è regolabile; per cui in definitiva all'uscita si ha un impulso posizionabile rispetto all'inizio della interrogazione Ly . Questo impulso costituisce la sonda da mandare nei circuiti di lettura della M., escludendo il mastro M6 mediante il deviatore Dv., ogni qual volta si desideri verificare i margini di funzionamento della M. stessa (vedi parte III^, 1.7.).

2. L'ACCUMULATORE E I REGISTRI DI MODIFICAZIONE

2.1. Generalità e schema logico elementare

2.1.1. - L'Accumulatore (A.) ed i Registri di Modificazione (T.) sono due organi aventi le caratteristiche funzionali della Memoria, però con capacità molto minore e compiti logici diversi.

- La ragione per cui A. e T. vengono trattati insieme risulta dal fatto che, non accadendo mai che questi organi debbano venire usati contemporaneamente, i circuiti che fisicamente li costituiscono sono unici, e vengono scambiati secondo l'occorrenza, in funzione di A. o di T.

- L'A. interviene durante la fase esecutiva delle operazioni aritmetiche. In A. infatti è immagazzinato uno dei due operandi da U.A. il quale viene estratto e presentato all'U.A. all'istante opportuno, contemporaneamente all'altro operando (proveniente da M.). Il risultato dell'operazione viene immagazzinato ancora in A., e costituisce il nuovo operando del successivo ciclo di calcolo, oppure, se è definitivo, viene successivamente trasferito in M.

- I registri T. intervengono nella fase di preparazione delle istruzioni, e servono a determinare l'indirizzo e lunghezza dell'operando da M. Questi infatti si ottengono sommando, in U.A., il contenuto di uno dei registri T (determinato dall'istruzione stessa) al l'indirizzo ed alla lunghezza contenute nell'istruzione. L'indirizzo così ottenuto è inviato al Registro I, e la lunghezza al Registro L .

- I Registri T possono anche funzionare come accumulatore, con lunghezza di parola fissa.

2.1.2. - Le informazioni sono operate in A. ed in T. a blocchi di 6 bits in parallelo. Di questi, 4 bit (a, b, c, d) costituiscono il carattere; uno (e) serve da controllo; l'ultimo (f) ha la funzione di indicare la fine di ogni parola contenuta in A. e T.

- L'A. consta di 100 celle (di 6 bit ciascuna), ognuna indirizzabile tramite i Registri V, dalla 0 alla 99 .
- I Registri T sono in numero di 10, ognuno composto di 10 celle, indirizzabili tramite i Registri V, dalla 0 alla 99 . Ogni volta che un indirizzo compreso tra i precedenti è presentato dai registri V, viene selezionata la cella corrispondente in A. oppure in T.; la scelta tra A. e T. è fatta tramite i comandi: selezione A e selezione T, che smistano l'indirizzo proveniente da V o in A o in T .
- Vi sono inoltre indirizzi fittizi, ai quali cioè non corrisponde la selezione di alcuna cella di A o di T . Un indirizzo è fittizio ogni volta che uno almeno dei registri V presenta la configurazione non decimale 0000 oppure 1000 .
- Una cella per volta o di A. o di T. è indirizzabile. Una informazione, sia in A. che in T., è immagazzinata (o estratta) serialmente, carattere per carattere, partendo da un indirizzo prescelto, e proseguendo con gli indirizzi ordinatamente crescenti.

2.1.3. - Dalla precedente esposizione risulta che il ciclo fondamentale di funzionamento dell'A. o del T. consiste nell'estrarre e/o immagazzinare un blocco di 6 bit in una cella prescelta.

- L'indirizzo della cella prescelta è presentato in modo continuo dai registri V agli organi di selezione dell'A.-T.; i comandi selezione A. e selezione T., smistano l'indirizzo in A. o in T.
- Un ciclo di operazione di A. o T. ha inizio quando si dà il comando di pilotaggio.
- Il ciclo è identico a quello che avviene in M.; consta cioè di due fasi, lettura o scrittura, succedentisi nel tempo. Come per la Memoria, il ciclo di lettura-scrittura si effettua sempre ed automaticamente una volta dato il comando di pilotaggio (a partire dal M 5).

LETTURA

Questa fase occupa l'A. (o il T) dal M 5 al M 7 (pi-

lotaggio scrittura).

- L'informazione da scrivere può provenire sia dall'esterno (U.A. o M.), sia dal canale di rigenerazione collegato con l'uscita S .
- Nel caso di scrittura attraverso il canale di rigenerazione, si immagazzina in A. (o T) il contenuto di A. (o di T.) (riscrittura).
- Ovviamente la scrittura dall'esterno esclude il canale di rigenerazione e viceversa; ossia dei due comandi: scrittura da U.A., riscrittura, uno solo per ciclo può essere presente.
- La fig. 2.1. rappresenta lo schema logico elementare dell'A.-T.; la fig.2.2. rappresenta il diagramma temporale di un ciclo di A. o di T.

2.2. Schema funzionale dell'A.-T.

2.2.1. - Abbiamo già stabilito che l'A. ed i registri T. hanno le stesse caratteristiche funzionali della Memoria (salvo la differenza di una capacità molto minore).

L'organo A.-T. è perciò sostanzialmente identico alla M., sia come principi fisici su cui si basa il funzionamento, sia come schema a blocchi.
Senza ripetere tutto quanto detto a proposito della M., ma riferendoci ad esso, ci basterà rilevarne le differenze per avere un quadro esatto del funzionamento dell'A.-T.

L'elemento fisico fondamentale è, come per la M., il nucleo di materiale magnetico, operato secondo i principi esposti nel paragr. 1.2.1. Valida è pure la disposizione in reticolo tridimensionale, mediante serializzazione di matrici a due dimensioni x e y .

Valido è pure il sistema di selezione mediante trasformatori a presa centrale, ordinati in sottomatrici (par. 1.2.3.); come pure il principio di lettura mediante discriminazione temporale (par. 1.2.4.).

Date le piccole dimensioni delle matrici di A.-T. non si è invece ritenuto sostanzialmente conveniente applicare il principio di anticoincidenza (paragrafo 1.2.2.); pertanto la selezione è effettuata attivando una sola linea x ed una sola linea y . Cade pertanto la necessità di avere due separati avvolgimenti di inibizione in una matrice (paragrafo 1.3.2.).

Infine, date sempre le piccole dimensioni delle matrici, non è necessario avere due avvolgimenti di lettura per matrice (par. 1.3.2.), e nemmeno sfalsare l'inizio degli impulsi di lettura x rispetto agli impulsi y (par. 1.3.3.), per migliorare il rapporto segnale/disturbo.

2.2.2. - L'organo centrale dell'A.-T., cioè il complesso dei nuclei magnetici, è costituito nel modo seguente:

nuclei totali: 1.200
matrici: 6, ciascuna di 200 nuclei (20 x 10)
fili di pilotaggio: 20 x e 10 y , comuni a tutte le matrici

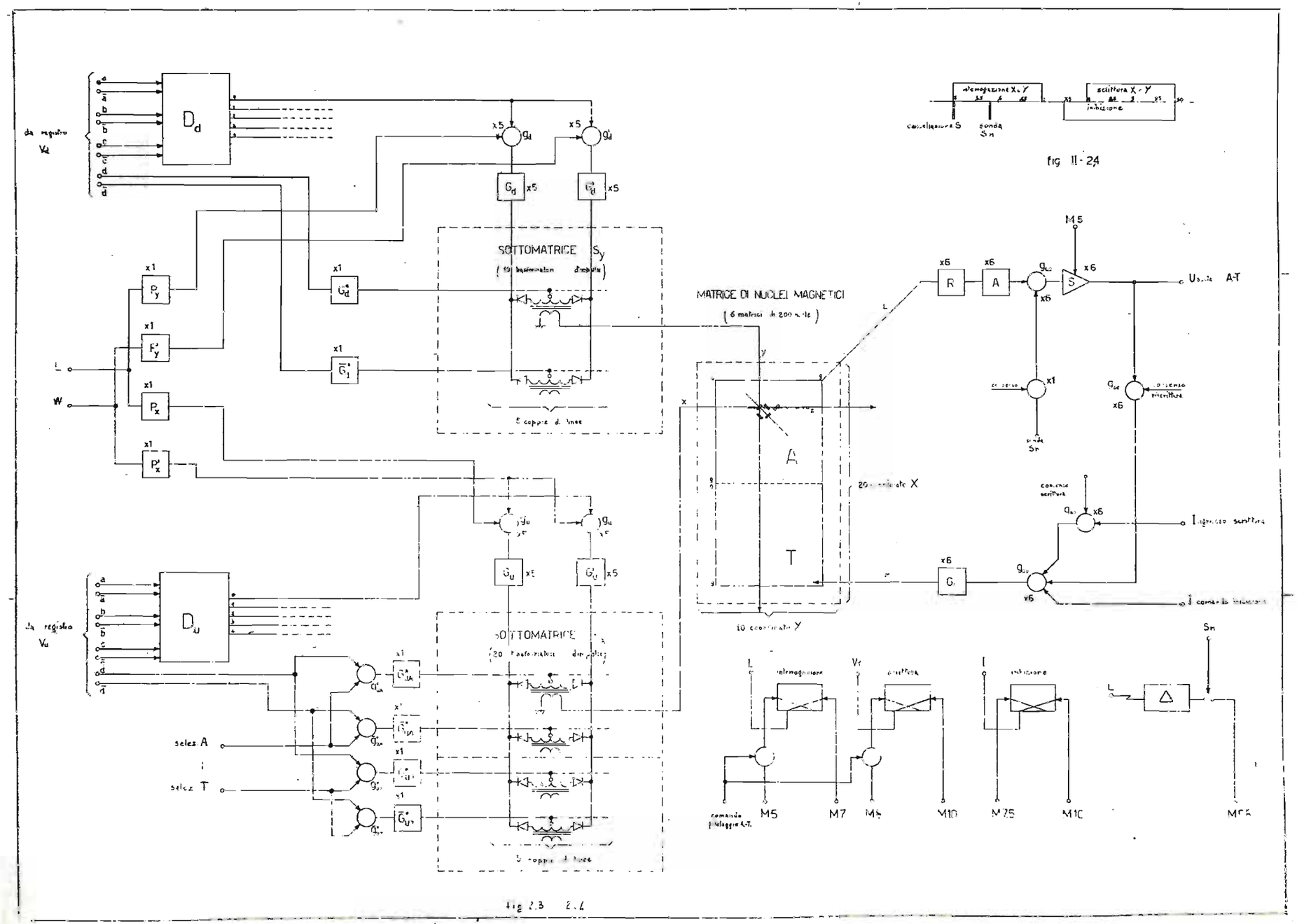


fig II-24

fig 2.3 2.4

fili di inibizione: 1 per matrice
fili di lettura: 1 per matrice.

Ogni matrice, costituente fisicamente un unico blocco, è funzionalmente divisa in due parti, ciascuna di 10 x 10 nuclei, corrispondenti rispettivamente all'A. ed al registro T.

In altre parole i fili di pilotaggio y sono comuni sia all'A. che al T., separati restando i fili di pilotaggio x (v. fig. 2.3.). Ciò è possibile per il fatto che, come già detto, non accade mai che l'A. ed il T. debbano venire selezionati contemporaneamente.

La disposizione adottata consente quindi una riduzione dell'onere complessivo del pilotaggio.

2.2.3. - La fig. 2.3. rappresenta lo schema funzionale a blocchi dell'A.-T. Nello schema noi possiamo distinguere gli stessi organi presenti nello schema funzionale (fig. 1.10.) della M. (v. par. 1.3.1.), cioè:

A) <u>Organi di decodificazione, selezione e pilotaggio.</u>
Partendo dai blocchi Vu e Vd, che rappresentano i due registri (unità, decine) contenenti l'indirizzo da selezionare, noi troviamo i due blocchi Du e Dd, che decodificano entrambi tre canali binari ($a\bar{a}$, $b\bar{b}$, $c\bar{c}$) in una su 5 linee.
Le linee decodificate servono alla selezione dei generatori come G e G' , collegati agli estremi dei trasformatori.
Invece i canali d , $\bar{d}$ vanno a selezionare i generatori come G^+ , collegati ai centri dei trasformatori; però mentre nella sottomatrice Sy i generatori G^+ sono direttamente selezionati dal canale d , $\bar{d}$ del registro Vd , nella sottomatrice Sx la selezione dei generatori G^+ è condizionata, tramite gli and gates g_u^+ , alla scelta tra A. e T.

Una volta effettuata la selezione, il pilotaggio viene comandato agli istanti prestabiliti, dagli ingressi L (lettura) e W (scrittura), tramite i piloti Px P'x e Py P'y e gli and gates g_u g'_u e g_d g'_d .

Lo schema di funzionamento è lo stesso di quello visto per la Memoria.

B) Organi di lettura. Sono rappresentati dal raddrizzatore ad onda intera R, dall'amplificatore a larga banda A, dal and gates gL1 (in cui avviene la discriminazione temporale tra segnale e disturbo) ed infine dallo staticizzatore temporaneo S , la cui uscita è, a tutti gli effetti, l'uscita di A.-T.

Il funzionamento dei blocchi sopra citati, è del tutto identico a quello dei blocchi analoghi dello schema 1.10. della M.

C) Organi di inibizione. Sono rappresentati dal generatore degli impulsi di inibizione G_I , col relativo and gate g_{s1} ; il gates g_{s2} stabilisce, su comando, il canale di rigenerazione. Come per la M., il gate g_{s2} e g_{s3} sono in opposizione di fase.

D) Forme d'onda di temporizzazione. Sono rappresentate nella fig. 2.4., e la loro interpretazione è immediata, per confronto con le analoghe (fig. 1.11.) della M.

Nella fig. 2.3., in angolo, sono disegnati i generatori delle forme d'onda di temporizzazione dell'A.T.

3. L'UNITA' ARITMETICA

3.1. Generalità

3.1.1. Scopi ed operazioni consentite

Compito fondamentale della U.A. è di eseguire tutte le operazioni aritmetiche richieste dal funzionamento della Macchina calcolatrice.
Le operazioni che possono essere eseguite sono:

- Somma
- Sottrazione
- Moltiplicazione additiva
- Moltiplicazione sottrattiva.

L'U.A. scambia segnali con le seguenti altre parti della macchina:

- Memoria
- Accumulatore
- Registri W , V , RL , O , I , Raus.P , Raus R .
- Nastro magnetico.

Fra le parti sopra elencate della macchina l'U.A. viene utilizzata anche come semplice mezzo di transito nelle operazioni di trasferimento.

3.1.2. Codice ed organo di somma

Il codice usato è di tipo biquinario, scelto per le peculiari caratteristiche di simmetria, che consentono notevole semplicità in molti circuiti.

Codice :

	a	b	c	d
0	1	0	0	0
1	1	1	0	0
2	1	1	1	0
3	0	1	1	0
4	0	0	1	0
5	1	0	0	1
6	1	1	0	1
7	1	1	1	1
8	0	1	1	1
9	0	0	1	1

Il segnale "d" rappresenta una cinquina; i segnali "a", "b", "c" definiscono i numeri da 0 a 4 .
L'organo fondamentale dell'U.A., che interviene in tutte le operazioni aritmetiche, è il Sommatore, costituito dalla Matrice di Somma e dall'Addizionatore Binario.
L'operazione eseguita dal Sommatore è la somma di due cifre decimali. Più specificatamente la Matrice di Somma opera sui segnali "a", "b", "c" delle cifre codificate fornendo in uscita i segnali "a", "b", "c" della cifra somma e l'eventuale segnale "r", che è una cinquina derivante dalla somma eseguita dei due numeri compresi fra 0 e 4 (dato che il segnale "d" non viene manipolato).
L'Addizionatore binario esegue binariamente la somma dei segnali "d" presenti in ingresso e del segnale di riporto "r", fornendo il segnale "d" del numero somma e l'eventuale segnale Rd di riporto decimale.
Tutte le operazioni eseguibili con l'U.A. sono ricondotte alla operazione di somma.

Matrice di somma

La Matrice di somma ha due ingressi che indicheremo con gli indici 1 e 2 . All'ingresso 1 normalmente giungono i segnali provenienti dalla memoria; all'ingresso 2 i segnali provenienti dall'Accumulatore.
La struttura logica della Matrice di Somma è realizzata in modo

tale da soddisfare quanto indicato nella tabella (1).
In questa tabella, nella prima riga sono riportati i segnali presenti all'ingresso 2, nella prima colonna i segnali presenti all'ingresso 1 . E' da notare che nella prima colonna la combinazione di segnali indicata rappresenta la decodificazione dei numeri da 0 a 4, così come è indicato nella finca adiacente.
In uscita si hanno i segnali indicati dalla intersezione fra ciascuna posizione di riga e colonna.

		a_A	b_A	c_A	$\bar{a}_A$	$\bar{b}_A$	$\bar{c}_A$	$c_A \cdot \bar{b}_A$
0	$a\ \bar{b}$	a_{so}	b_{so}	c_{so}				
1	$a\ b\ \bar{c}$	b_{so}	c_{so}			a_{so}	a_{so}	r
2	a c	c_{so}			a_{so},r	a_{so}		
3	$\bar{a}\ b$			a_{so},r	b_{so}	b_{so},c_{so}	c_{so}	
4	$\bar{b}\ c$		a_{so},r	b_{so},r	c_{so}	c_{so}		

Tabella 1

L'addizionatore binario ha tre ingressi d_1 , d_2 , r e funziona secondo quanto risulta dalla tabella (II).

ingressi / risultato	1	2	3
	d_M	d_A	r
si ha riporto decimale R_d quando:	1	1	0
	1	0	1
	0	1	1
si ha riporto cinquinale d_{so} quando:	1	0	0
	0	1	0
	0	0	1
	1	1	1

tabella (II)

3.1.3. Organi principali dell'U.A.

I canali principali attraverso cui corrono i segnali nella U.A. sono quelli eseguiti in grassetto nello schema a blocchi di figura 3.1. I segnali provenienti dalla Memoria vengono trasferiti in accumulatore, senza essere operati, tramite la porta GMA . Quando occorre operarli aritmeticamente essi, tramite la porta GMU, passano attraverso l'organo Complementatore ed il successivo organo Aggiungitore di unità. I segnali uscenti dall'aggiungitore di unità entrano nella Matrice di Somma e nell'addizionatore binario attraverso l'ingresso 1 .
I segnali provenienti dall'accumulatore passano sempre attraverso un organo Complementatore ed un successivo organo aggiungitore di unità. I segnali uscenti dall'aggiungitore di unità possono essere inviati direttamente alla Memoria tramite la porta GAM e giungono all'ingresso 2 della Matrice di Somma e dell'addizionatore binario tramite la porta controllata da FO1 .
Gli organi rappresentati all'interno del rettangolo tratteggiato intervengono nel funzionamento dell'U.A. soltanto durante l'operazione di moltiplicazione.
La loro funzione sarà meglio illustrata in un successivo capitolo.

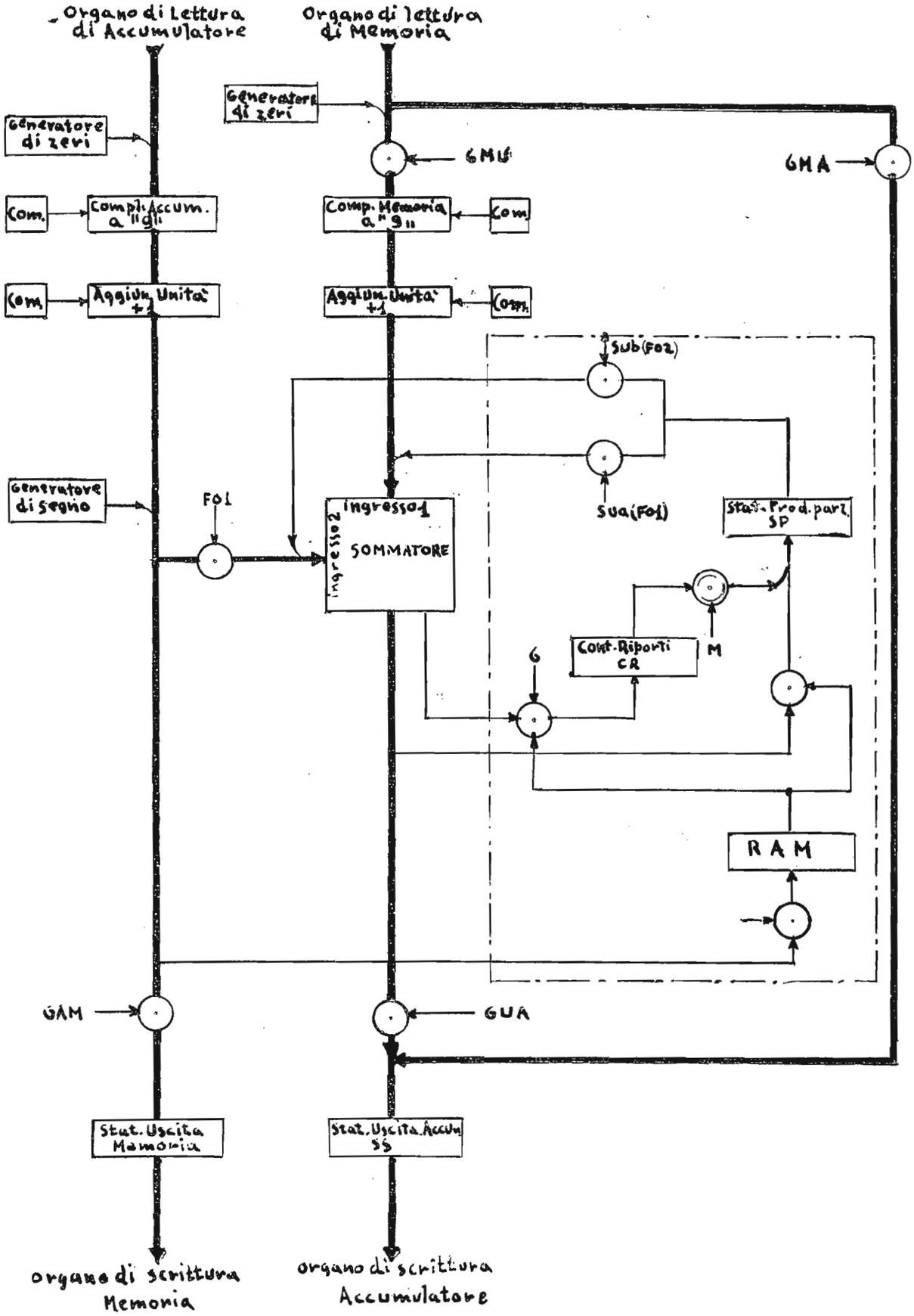

Organo di Lettura di Accumulatore
Organo di lettura di Memoria
Generatore di zeri
Generatore di zeri
GMU
GMA
Com.
Compl. Accum. a "9"
Comp. Memoria a "9"
Com.
Com.
Aggiun. Unità +1
Aggiun. Unità +1
Com.
Sub(F02)
Sua(F01)
Stat. Prod. parz. SP
Generatore di segno
F01
ingresso1
ingresso2
SOMMATORE
Cont. Riporti CR
M
G
RAM
GAM
GUA
Stat. Uscita Memoria
Stat. Uscita Accum. SS
organo di scrittura Memoria
organo di scrittura Accumulatore

Fig. 3-1

3.2. Descrizione aritmetica delle operazioni con riferimento alle possibilità funzionali delle macchine.

3.2.1. Somma di due numeri positivi

I due numeri da sommare sono contenuti uno in Memoria (M) ed uno in accumulatore (A_o). La somma viene eseguita cifra per cifra, cominciando dalla meno significativa, ed il risultato A_1 si scrive in accumulatore nelle celle inizialmente occupate da A_o .

Siano

$$M = 10^{p-1} m_p + \dots + 10 m_2 + m_1$$

$$A_o = 10^{q-1} a_{q.0} + \dots + 10 a_{2.0} + a_{1.0}$$

Nel primo periodo di cifra (pdc) della fase esecutiva β la cifra m_1 si presenta all'ingresso 1 del sommatore e la cifra $a_{1.0}$ allo ingresso 2 .

In uscita del sommatore si ha la cifra $a_{1.1}$ e l'eventuale riporto di decina R_1 . La cifra $a_{1.1}$ viene scritta in accumulatore al posto di $a_{1.0}$, mentre il riporto R_1 va ad azionare l'organo "aggiungitore di unità" sul canale di Memoria.

Nel secondo pdc la cifra M_2 proveniente da Memoria, passando attraverso l'aggiungitore di unità (sempre che ci fosse stato il riporto R_1), si trasforma nella cifra $M_2^1 = M_2 + 1$ e come tale si presenta all'ingresso 1 del Sommatore. Contemporaneamente all'ingresso 2 si presenta la cifra $a_{2.0}$ ed in uscita del Sommatore si ha la cifra $a_{2.1}$ e l'eventuale riporto di decina R_2 . La cifra $a_{2.1}$ si scrive in Accumulatore al posto di $a_{2.0}$ ed R_2 va ad azionare l'aggiungitore di unità come nel caso precedente.

L'operazione prosegue secondo queste modalità fino al pdc in cui compare il segnale $\emptyset_c$ indicante che è stata letta l'ultima cifra del numero in Memoria.

Se in questo pdc non vi è riporto R_p , l'operazione termina e compare il segnale φ' di fine operazione.

Se invece vi è il riporto R_p, l'operazione continua, ed al posto delle cifre di memoria compaiono degli zeri all'ingresso del canale di memoria, generati da apposito organo.
Il segnale φ' comparirà nel primo pdc in cui mancherà il riporto R_d : al massimo potrà essere il pdc successivo a quello in cui compare il segnale f_A , indicante che è stata letta l'ultima cifra del numero in Accumulatore (Vedi Circuiti di fine).

3.2.2. I segni

I numeri in Memoria sono sempre espressi in valore assoluto e segno salvo il caso richiamato più avanti; se il segno non è esplicitamente indicato si intende che il numero sia positivo.

I numeri in Accumulatore possono essere espressi sia sotto forma di valore assoluto, sia sotto la forma di complemento (*), con segno o senza segno; hanno segno soltanto quando almeno uno qualunque dei numeri operandi è segnato. Durante il trasferimento di un numero da Accumulatore a Memoria, se è in complemento con segno viene trasformato in valore assoluto e segno mediante complementazione, se invece è in complemento senza segno, viene trasferito senza complementazione: è questo l'unico caso in cui in memoria i numeri sono sotto forma di complemento.

La logica della macchina è tale che il segno dell'accumulatore non cambia mai durante le operazioni aritmetiche: esso è sempre positivo fin tanto che non avviene il trasferimento di un numero negativo da Memoria in Accumulatore.

Nell'esecuzione di operazioni aritmetiche occorre prendere in considerazione i seguenti segni:

S_{op} = segno dell'operazione.

S_M = segno del numero in Memoria.

+/- = segno del numero in Accumulatore.

S_R = segno del numero moltiplicatore (in moltiplicazione).

Il prodotto algebrico del Segni $S_{op} \times S_M \times S_R = \pm$ determina se il numero proveniente della Memoria, eventualmente moltiplicato per il numero moltiplicatore, debba essere sommato ($\pm = +$) o sottratto ($\pm = -$) al numero contenuto nell'Accumulatore.

Siccome il numero risultante $A_1 = \pm M + A_o$ ha sempre il segno di A_o, si vede che:

(*) - Si ricorda che per complemento di un numero $A = 10^{p-1} a + \ldots$ $\ldots + 10^{i-1} a_i + \ldots + a_1$ si intende il numero che si ottiene ponendo al posto della prima cifra significativa, diversa da zero, a_i la cifra $(10-a_i)$, ed al posto di tutte le altre cifre che seguono a_i, la cifra $(9 - a_{i+1})$, $(9 - a_{i+2})$, etc. Si dice che a_i è complementata "a 10" e che le successive sono complementate "a 9". Gli zeri che precedono a_i rimangono inalterati.

se +/- = + $\quad A_1 = \pm M + (+A_o) = +(\pm M + A_o)$ ossia $+.\pm = \pm$

se +/- = - $\quad A_1 = \pm M + (-A_o) = -(\mp M + A_o)$ $\quad -.\pm = \mp$

Questo dice che il segno risultante $S_{op} \times S_M \times S_R$ va algebricamente moltiplicato per il segno dell'accumulatore:
L'operazione $- M + A_o = A_1$ si esegue nello stesso modo esposto per la somma di due numeri positivi, pur di complementare il numero M prima di introdurlo all'ingresso 1 del Sommatore, secondo quanto esposto nel successivo paragrafo.

3.2.3. Sottrazione

L'operazione $- M + A_o$ può scriversi nella forma:

$$- M + A_o + h.10^k - h.10^k = (h.10^k - M) + A_o - h.10^k$$

Se $\quad M = 10^{p-1} m_p + \dots + m_1$

$$A_o = 10^{q-1} a_q + \dots + a_1$$

h e k siano due numeri positivi arbitrari, purchè si scelga k sempre maggiore del più grande fra i numeri p e q .
A meno del termine $- h.10^k$, l'operazione $- M + A_o$ si può ridurre alle somme dei due numeri positivi $(h.10^k - M) + A_o$.
Posto h = 1 . Si può anche scrivere

$$(h.10^k - M) = 10^{k-1}.9 + 10^{k-2}.9 + \dots + 10^{p-1}(9 - m_p) + \dots +$$

$$+ 10(9 - m_2) + (10 - m_1)$$

per cui, l'addendo da sommare ad A_o non è altro che il Complemento del numero contenuto in Memoria esteso, dalla parte della cifra più significativa, di k - p zeri.

L'operazione si esegue pertanto nel modo seguente.
La presenza del segno risultante - nel prodotto dei segni $S_{op} . S_M . S_R$. +/- aziona il Complementatore "a 9" sul canale di Memoria, che rimane attivo per tutta la durata dell'operazione.
L'aggiungitore di unità sul canale di Memoria aggiunge "1" a tutte le cifre provenienti dal complementatore fino al pdc, questo

compreso, in cui si legge in Memoria la prima cifra meno significativa diversa da zero.
In questo modo all'ingresso 1 del Sommatore si presentano successivamente le cifre $(10 - m_1)$, $(9 - m_2)$, $(9 - m_3)$ etc. e si sommano con le corrispondenti cifre a_1 , a_2 , a_3 etc. provenienti dall'accumulatore secondo le modalità esposte al paragrafo 3.2.1.

Dopo la comparsa del segnale $\emptyset_c$, in uscita del complementatore si hanno sempre dei 9 originati dalla complementazione degli zeri introdotti in ingresso del canale di Memoria.

L'operazione termina, dopo la comparsa di $\emptyset_c$, nel pdc in cui si ottiene il primo riporto decimale (Infatti nei successivi pdc le cifre in ingresso 1 del Sommatore sarebbero sempre 9 + 1 = 0 per cui le cifre di accumulatori corrispondenti rimarrebbero inalterate).

Il numero contenuto nell'accumulatore alla fine dell'operazione può essere il valore assoluto o il complemento del risultato secondo che era $M < A_o$ oppure $M > A_o$, secondo quanto esposto nel paragrafo successivo.

3.2.4. Il contenuto dell'Accumulatore

Il numero contenuto in Accumulatore è individuato da tre elementi:

1) Norma = numero effettivamente contenuto nelle celle di accumulatore.

2) Segno +/- = segno di positività o negatività conservato in apposito organo.

3) C/N = indicazione Normale (N) o Complemento (C) secondo che la norma è il valore assoluto o il complemento del numero.

La Norma è il numero che viene operato nel Sommatore durante le operazioni aritmetiche ed è un numero positivo.
Se essa rappresenta il valore assoluto (C/N = N) le cifre che seguono quelle più significative sono degli zeri e vengono introdotti da apposito generatore di zeri all'ingresso del canale di accumulatore quando l'operazione debba proseguire oltre il pdc in cui compare il segnale f_A .
Se essa rappresenta il complemento (C/N = C), le cifre che seguono quella più significativa sono dei 9, e si ottengono per mezzo dello stesso generatore di zeri azionando però il complementatore "a 9" nel canale di Accumulatore per tutti i pdc che

seguono quello in cui è comparso il segnale f_A , fino alla fine dell'operazione.

L'indicazione di normalità o complementarità della Norma A_1 si ottiene nel pdc in cui compare il segnale di fine operazione φ', ed è funzione del

- segno complessivo di operazione = $S_{op} \cdot S_M \cdot S_R \cdot +/-$,
- dello dello stato di C/N relativo alla Norma di A_o ,
- della presenza (1) o assenza (0) del riporto decimale R nel pdc in cui compare il segnale φ' .

I casi che possono scaturire dall'operazione $\pm M + A_o = A_1$ sono raggruppati nella seguente tabella:

Caso	Segno risultante $S_{op} \cdot S_M S_R \cdot +/-$	$(C/N)_{A_o}$	$R_{\varphi'}$	$(C/N)_{A_1}$
1	+	N	1	N
			0	N
2	+	C	1	N
			0	C
3	-	N	1	N
			0	C
4	-	C	1	C
			0	C

Tabella 3

Infatti se - p - indica il numero di cifre del numero M proveniente da Memoria e - q - indica il numero di cifre del numero A_o proveniente da accumulatore il risultato corretto nei quattro casi sarebbe:

1 - $+ M + A_o = A_1$ = numero positivo quindi normale con o senza $R_{\varphi'}$.

2 - $+ M + (A_o - 10^q) =$

$= (+M + A_o) - 10^q =$ numero positivo soltanto se vi è ripetuto R_{φ}, nell'operazione $M + A_o$ effettuata dalla macchina perchè l'operazione procede fino a che è stato letto il più lungo dei numeri M, A_o (v. circuiti di fine).

3 - $(+M-10^p) + A_o =$
$=(+M+A_o) - 10^p =$
$= A_1 - 10^p$

numero positivo soltanto se vi è riporto R_{φ}, nell'operazione $M + A_o$ effettuata dalla macchina, perchè l'operazione procede sempre fino a che è stata letta l'ultima cifra di memoria.

4 - $(+M-10^p)+(A_o-10^q) =$
$=(+M+A_o)-(10^p+10^q)=$
$= A_1 - (10^p + 10^q)$

numero sempre negativo quindi il risultato $M + A_o$ è il suo complemento sempre, con o senza R_{φ}.

Nei casi 2 e 3, se non vi è riporto, il numero in accumulatore A_1 è il complemento del risultato dell'operazione.

3.2.5. Moltiplicazione additiva

La moltiplicazione si dice additiva quando il segno complessivo dell'operazione $S_{op} \cdot S_M \cdot S_R \cdot +/-$ confrontato con la indicazione di normalità o complementarità della cifra di Moltiplicatore (v. prf. 3.2.6.) è positivo.

La moltiplicazione si esegue mediante un procedimento di somme ripetute.

L'operazione è del tipo $M \times T + A_o$ in cui

M è il moltiplicando, composto delle cifre $m_p, \ldots m_2, m_1$ (m_1 cifra meno significativa)

T è il moltiplicatore, composto dalle cifre $t_r, \ldots t_2, t_1$ (t_1 cifra meno signif.)

A_o è l'addendo, composto dalle cifre $a_q, \ldots a_2, a_1$ (a_1, cifra meno significativa).

M è un numero contenuto in memoria, T è un numero contenuto nel registro T_o, A_o è un numero contenuto in accumulatore. Tutti e tre sono numeri positivi: il loro segno, rispettivamente S_M, S_R, $+/-$ è stato considerato separatamente, per decidere del segno complessivo dell'operazione.

Scrivendo l'operazione così

$$M \times T + A_o = A_o + M.t_1 + 10\ M.t_2 + 10^2\ M.t_3 + \dots + 10^{r-1}\ M.t_r \quad ,$$

si vede che essa si riduce alla somma di successivi termini del tipo

$$A_{h-1} + 10^{h-1} \cdot M.t_h = A_h \qquad (1)$$

Infatti:

$$\begin{array}{r} A_o + M.t_1 = A_1 \\ A_1 + 10\ M.t_2 = A_2 \\ \hline A_{r-1} + 10^{r-1}\ M.t_r = A_r \end{array}$$

Per effetto del coefficiente 10^{h-1} di cui è affetto il numero $M.t_h$, le prime h-1 cifre del numero A_{h-1} non intervengono nella somma (1), e rappresentano già le corrispondenti cifre del risultato finale.

Ciò permette di considerare come operazione tipo la seguente

$$A'_h + M.t_h = A''_h \qquad (2)$$

dove

$$A'_h = A''_{h-1}$$

quando in questo numero si trascuri la cifra meno significativa. Quando h = 1 ,

$$A'_h = A_o$$

Il risultato finale è il numero A''_r a cui si affianchino, dalla parte della cifra meno significativa, le cifre scartate di A''_{r-1} , A''_{r-2} , ... , A''_1 , nell'ordine.

Es. M = 12548 T = 572 A_o = 1356

$A'_1 + M.t_1$ = 1356 + 12548 x 2 = 2 6 4 5 2 = A''_1

$A'_2 + M.t_2$ = 2645 + 12548 x 7 = 9 8 4 8 1 = A''_2

$A'_3 + M.t_3$ = 9848 + 12548 x 5 = 7 1 7 8 8 = A''_3 (r = 3)

$$A_o + M.T = 7178812$$

L'esecuzione dell'operazione (2) costituisce una sottofase di moltiplicazione; le sottofasi di moltiplicazione si susseguono fino ad operare tutte le cifre t_h del Moltiplicatore.
Il risultato della prima sottofase , A''_1 , è scritto nell'accumulatore al posto di A_o ; il risultato della seconda, A''_2 , è scritto nell'accumulatore si trova così scritto il risultato finale.

Durante una sottofase di moltiplicazione, in ciascun pdc, si esegue una operazione del tipo:

$$r_{k-1} + m_k.t_h + a_k \qquad (3)$$

Infatti, l'operazione $A'_h + M.t_h$ si può scomporre nel seguente modo. Sia:

$$A'_h = 10^{s-1} \; a'_{s,h} + \dots + 10 \; a'_{2,h} + a'_{1,h}$$

si può scrivere

$$A'_h + M.t_h = (a'_{1,h} + m_1.t_h) + 10(a'_{2,h} + m_2.t_h) + 10^2(a'_{3,h} + m_3.t_h) + \dots$$

Il numero risultante dell'operazione indicata fra le parentesi del primo addendo si può esprimere mediante la somma della sua cifra meno significativa $a''_{1,h}$ e del riporto di decina $10.r_{1,h}$. Analogamente si può fare per i successivi termini.

$$\begin{aligned} (a'_{1,h} + m_1.t_h) &= 10 \; r_{1,h} + a''_{1,h} \\ 10 \; (r_{1,h} + a'_{2,h} + m_2.t_h) &= 10^2 \; r_{2,h} + 10 \; a''_{2,h} \\ 10^2(r_{2,h} + a'_{3,h} + m_3.t_h) &= 10^3 \; r_{3,h} + 10^2 \; a''_{3,h} \end{aligned} \qquad (4)$$

Sostituendo si ha

$$A'_h + M.t_h = a''_{1,h} + 10 \; a''_{2,h} + 10^2 \; a''_{3,h} + \dots$$

ed il risultato dell'operazione è dato dalla successione delle cifre

$$A''_h = \dots \; a''_{3,h} \; , \; a''_{2,h} \; , \; a''_{1,h}$$

L'esecuzione dell'operazione indicata fra parentesi nella (4) avviene in 1 pdc ; in ciascun pdc l'operazione è quindi del tipo (3). Nel periodo k-esimo r_{k-1} è il riporto del periodo precedente k-1 esimo.

Il prodotto $m_k . t_h$ è ottenuto sommando t_h volte a se stessa la cifra m_k , cosicchè in macchina si eseguono soltanto somme:

$$r_{k-1} + m_{k_1} + m_{k_2} + \dots + m_{k_n} + a_k$$

Per ridurre il numero delle somme nel pdc si è introdotto il complemento "a 10" della cifra di moltiplicatore quando questa è $\geqslant 5$. Per il caso $t_h = 5$ è aritmeticamente indifferente complementare o no, ma la scelta dipende da considerazioni funzionali.
Quando la cifra t_h è $\geqslant 5$ l'operazione assume la seguente forma.
Sia

$$A_o + M.T = A_o + M.t_1 + 10\ M.t_2 + \dots + 10^{h-1}\ M.t_h +$$

$$+ 10^h\ M.t_{h+1} + \dots$$

posto $$t_h = -(10 - t_h) + 10$$

sostituendo si ha

$$A_o + M.T = A_o + M.t_1 + 10\ M.t_2 + \dots - 10^{h-1}.M.(10 - t_h) +$$

$$+ 10^h\ M\ (t_{h+1} + 1) + \dots$$

L'introduzione del complemento della cifra t_h cambia il segno complessivo di operazione della sottofase di moltiplicazione h-esima $A_{h-1} - M.(10 - t_h)$ ed obbliga ad aumentare di una unità la successiva cifra di moltiplicatore t_{h+1} .

Il primo pdc di ciascuna sottofase di moltiplicazione è utilizzato per trasferire la cifra di moltiplicatore operanda dal registro T_o al registro attuale di moltiplicazione (RAM); durante il trasferimento si esegue l'eventuale complementazione o l'aggiunta di una unità.

Quando la cifra <u>più significativa del moltiplicatore</u> t_r è $\geqslant 5$ si rende necessario considerare ancora una cifra successiva, uguale a zero, a cui si aggiunge l'unità nel primo pdc di una ulteriore (r+1 esima) sottofase di moltiplicazione:

$$A_o + M.T = A_o + M.t_1 + \ldots - 10^{r-1}\ M.(10 - t_r) + 10^r\ M.(0 + 1)$$

Durante un pdc la macchina esegue le somme dei termini

$$r_{k-1} + m_{k_1} + m_{k_2} + \ldots + m_{k_i} \qquad \text{con} \quad i \leqslant 5\ ,$$

dal µs 0,5 al µs 4,5 con la cadenza di una somma per microsecondo. La somma con l'addendo a_k avviene al µs 7 dello stesso pdc secondo le modalità normali dell'addizione, poichè a_k si presenta all'ingresso 2 del Sommatore mentre la cifra meno significativa del parziale $(r_{k-1} + m_{k1} + m_{k2} + \ldots + m_{ki})$ si presenta all'ingresso 1 . Gli eventuali riporti che nascono dal µs 0,5 al µs 4,5 o al µs 7 vengono contati e trattenuti in apposito organo (Contatore dei Riporti, v. prf. 3.3.2.).
La cifra di moltiplicando operata nel pdc k esimo viene effettivamente letta in memoria al µs 6 del pdc precedente k-1 esimo e la sua presenza si protrae fino al µs 5 del pdc k-esimo.

3.2.6. Moltiplicazione sottrattiva

L'indicazione del fatto che la cifra di moltiplicatore t_h è $\geqslant 5$ è conservata in apposito organo, e viene confrontata con il segno complessivo di operazione $S_{op.} \cdot S_M \cdot S_R \cdot +/-$ trattandola come fosse un segno - se $\geqslant 5$ e un segno + se < 5 (v. prf. 3.2.5.): se i due segno sono uguali l'operazione è additiva, se sono discordi è sottrattiva.
La moltiplicazione sottrattiva $A_o - M.T$ si riconduce immediatamente a quella additiva scrivendola in questo modo:

$$A_o - MT + T.10^k - T.10^k = A_o + (10^k - M).T - T.10^k$$

A meno del termine sottrattivo $- T.10^k$ il risultato è quello di una operazione additiva in cui il numero proveniente da memoria viene complementato.

Siano p , r , q il numero delle cifre rispettivamente di M , T , A_o , il numero k deve essere un numero positivo arbitrario, maggiore del più grande fra i due numeri p+r e q .

Da quanto è stato detto nei prff. 3.2.5. e 3.2.6. si vede che la moltiplicazione si riduce sostanzialmente all'operazione di somma, cifra per cifra, del numero $M.t_h$ (oppure $(10^k - M).t_h$) che si presenta all'ingresso 1 del Sommatore e del numero A'_h che si presenta all'ingresso 2 .

Passando da una sottofase di moltiplicazione alla successiva si passa da t_h a t_{h+1} ed in accumulatore si sposta di una cella verso la cifra più significativa la lettura della prima cifra operanda.

3.2.7. La fine delle operazioni

3.2.7.1. Somma e Sottrazione

Il segnale di fine operazione φ' può comparire soltanto nel pdc in cui compare il segnale di fine memoria $\emptyset_c$ o nei successivi pdc .
Dopo la comparsa di $\emptyset_c$ le cifre uscenti dal complamentatore sul canale di memoria sono degli zeri o dei nove secondo che il segno complessivo dell'operazione è + o - , ossia secondo che non vi è complementazione a nove o vi è complementazione a nove degli zeri che entrano nel canale di memoria.

L'operazione s'arresta nel pdc in cui, mancando un riporto di decina - nel primo caso - od essendoci un riporto di decina - nel secondo caso - in ingresso 1 del Sommatore si presentassero successivamente sempre degli zeri che lascerebbero invariate le corrispondenti cifre dell'accumulatore.
La complementazione a nove è determinata dalla presenza del segnale D .

D	Riporto	φ'
0	1	0
	0	1
1	1	1
	0	0

Se non compare prima, il segnale φ' , compare nel pdc che segue quello in cui si legge l'ultima cifra di accumulatore. Nei casi in cui D = 0 e Rip. = 1 oppure D = 1 e Rip.=0 l'operatore prosegue fino alla lettura dell'ultima cifra di accumulatore, contemporaneamente alla quale compare il segnale f_A di fine accumulatore. La fine di operazione interviene in questo stesso pdc o nel successivo, secondo le modalità espresse dai circuiti di fine.

3.2.7.2. In moltiplicazione

In moltiplicazione il segnale di fine φ' può comparire soltanto nel pdc in cui compare il segnale $\emptyset_M$ o nei pdc successivi.
Il segnale $\emptyset_M$ compare nel secondo pdc successivo a quello in cui è comparso il segnale $\emptyset_c$ contemporaneo alla lettura dell'ultima cifra in Memoria; indicheremo con p questo periodo di cifra.
Nel pdc p+1 viene operata la cifra di memoria letta in p , e soltanto nel pdc successivo p+2 è possibile stabilire se l'operazione deve terminare o proseguire, qualunque sia la moltiplicazione eseguita. In qualche caso sarebbe possibile decidere anche nel pdc p+1 , ma per semplicità circuitale si è introdotto il ritardo fisso di due pdc rispetto alla comparsa di $\emptyset_c$.

a - Moltiplicazione additiva
Il numero proveniente da memoria operato al pdc p+2 è uno zero per cui l'operazione è $r_{p+1} + 0.t_h + a_{p+2}$, cioè una somma normale.
Il segnale φ' può comparire se non vi è riporto decimale, altrimenti l'operazione continua, come detto per il caso della Somma.

b - Moltiplicazione sottrattiva
La cifra di moltiplicando operata nel pdc p+2 è un 9 derivante dalla complementazione "a 9" dello zero introdotto nel canale di memoria del generatore di zeri dopo la comparsa di $\emptyset_c$, e l'operazione è

$$r_{p+1} + 9.t_h + a_{p+2}$$

Il riporto parziale $(r_{p+2})'$ originato dall'operazione

$r_{p+1} + 9.t_h$ può essere $r'_{p+2} = t_h - 1$

oppure $r'_{p+2} = t_h$

Se $r'_{p+2} = t_h - 1$ l'operazione termina se al μs 7 vi è un riporto di somma R = 1 perchè sarebbe $r_{p+2} = r'_{p+2} + 1 = t_h$, e nel pdc successivo l'operazione $t_h + 9t_h + a_{p+3} = 10\ t_h + a_{p+3}$ lascia inalterata la cifra di accumulatore ed il riporto t_h si propagherebbe in tutti i successivi pdc .
Se il riporto di somma R manca, l'operazione continua fino al pdc in cui compare per la prima volta.
Il caso $r'_{p+2} = t_h$ è anomalo, perchè, pur potendo l'operazione terminare, il circuito è tale che non essendoci riporto

di somma R , il circuito di fine non è attivato e quindi l'operazione prosegue.

Se non intervengono condizioni che fanno terminare l'operazione prima, il segnale φ' compare nel pdc successivo a quello in cui si legge l'ultima cifra del numero in accumulatore, come detto nel caso della Somma e Sottrazione.

3.3. Descrizione della esecuzione delle operazioni, con riferimento ai tempi ed agli organi interessati.

3.3.1. L'operazione di somma e sottrazione

La descrizione che segue riguarda la fase esecutiva dell'operazione: la fase preparatoria α è descritta al cap. 5 della parte II^.
L'operazione eseguita è: $\pm M + A$

ove $M = m_p , \ldots m_3 , m_2 , m_1$ è il numero in memoria

ed $A = a_q , \ldots a_3 , a_2 , a_1$ è il numero in accumulatore.

A destra della prima cifra di memoria m_1 può esistere una cifra indicante il segno del numero M .
Nel primo pdc della fase β viene letto in memoria il contenuto della prima cella selezionata che contiene il segno S_M , se il numero ha un segno, oppure la prima cifra m_1 del numero operando. L'indicazione del contenuto è presente all'ingresso del canale di memoria dell'U.A. del µs 6 fino al µs 5 del pdc seguente.

Contemporaneamente viene letto in Accumulatore il contenuto della prima cella selezionata che contiene la cifra a_1 del numero operando . L'indicazione del contenuto è presente all'ingresso del canale di accumulatore dell'U.A. del µs 6 fino al µs 5 del pdc successivo.

Nella fase preparatoria α è stato deciso, dal confronto dei segni (v. 3.2.2.), se il numero M deve essere complementato "a 9" o no ed è corrispondentemente presente o seguente il segnale D che consente tale complementazione.
La presenza del segnale D consente anche all'aggiungitore di unità di essere efficace nel primo pdc della fase β ed in tutti i successivi, fino al pdc, questo incluso, in cui compare la prima cifra significativa diversa da zero del numero in memoria (v.3.4.1.).

Il segnale D è presente per tutta la durata dell'operazione; il segnale +1, che consente all'aggiungitore di unità di essere efficace, è presente in tutti i pdc sopra ricordati dal µs 3 al µs 7,5 (flip-flop).
I segnali provenienti dalla Memoria e dall'Accumulatore sono utilizzati dall'U.A. dal µs 5 al µs 9 mediante la forma d'onda F01 per tutte le operazioni, esclusa la moltiplicazione, per la quale il segnale da Memoria è utilizzato da µs 9 al µs 5 mediante la

forma d'onda F02 . La F01 (o F02) controlla la porta GMUA per cui il segnale in transito da GMUA sarà presente dal μs 6 al μs 9 (coincidenza fra F01 ed il segnale da memoria).
La stessa durata (6 ÷ 9) avrà il segnale in transito dalla porta GAS_2 attraverso cui il segnale dell'accumulatore entra nel sommatore (ingresso 2).
Dal μs 6 al μs 7 i segnali della Memoria, eventualmente modificati dal complementatore e dall'aggiungitore di unità, ed i segnali dell'accumulatore si combinano nel Sommatore. Al μs 7 i segnali in uscita del Sommatore sono a regime e, tramite la porta GUA , vengono scritti nello staticizzatore di uscita S.S.
Al μs 7,5 la cifra contenuta in S.S. viene scritta in Accumulatore nella cella occupata da a_1 .

Al μs 7,5 cessa di esistere l'ordine +1 per cui, da questo istante, le uscite dal Sommatore non sono più corrette, se tale organo era efficace. L'eventuale presenza del segnale R di riporto decimale all'uscita del sommatore viene ricordata mediante uno staticizzatore dal μs 7 al μs 4 del pdc successivo.
Al μs 3 il contenuto di questo staticizzatore determina la comparsa dell'ordine +1 la cui efficacia si protrae dal μs 6 al μs 7,5 dello stesso periodo.

Se nel primo pdc la cifra proveniente da memoria è un segno S_M , l'operazione eseguita dal sommatore è priva di significato, la porta GUA rimane chiusa ed il risultato non viene scritto nello staticizzatore S.S.; l'eventuale segnale di riporto R è bloccato dalla porta GR .

Nel secondo pdc la cifra letta in memoria sarà m_1 se la precedente era S_M , oppure m_2 . La cifra letta in accumulatore sarà ancora a_1 nel primo caso, sarà invece a_2 nel secondo. Se nel primo pdc si era letto S_M , la cifra a_1 letta nella prima cella dell'accumulatore vi è riscritta, ed i registri V mantengono inalterato per tutto il periodo l'indirizzo di tale cella.

Dal μs 6 al μs 7 del secondo pdc la cifra proveniente da memoria, eventualmente modificata dal complementatore e dall'aggiungitore di unità, si combina nel Sommatore con la cifra proveniente dall'accumulatore. Al μs 7 il risultato viene scritto negli staticizzatori di uscita, ed al μs 7,5 il contenuto di questi è scritto nella cella di accumulatore occupata dalla cifra su cui si è operato.

L'operazione continua fino al pdc in cui compare il segnale di fine operazione φ '. Se l'operazione deve continuare dopo che si è letta in memoria l'ultima cifra, occorre introdurre sul canale di memoria la cifra zero per ogni pdc successivo, cioè a partire da quello in cui compare $Ø_{M_1}$, simulando in tal modo l'esistenza di zeri a sinistra della cifra più significativa del numero ope-

rato per tutti i pdc necessari al compimento dell'operazione. Siccome dal pdc in cui compare $\emptyset_{M1}$ i segnali in uscita dalla unità di lettura della memoria sono permanentemente nulli, sul filo del bit a si invia un segnale - lo stesso $\emptyset_{M1}$ - che trasforma lo zero assoluto (a = 0 , b = 0 , c = 0 , d = 0) in uno zero codice (a = 1 , b = 0 , c = 0 , d = 0).

Analogo provvedimento permette di scrivere uno zero codice nel canale di accumulatore nei pdc successivi a quello in cui compare il segnale $\emptyset_A$ indicante che si è letta in accumul. l'ultima cifra del numero operato, quando l'operazione deve continuare.

Se il numero in accumulatore è un complemento (C/N = C), gli zeri che vengono introdotti dopo $\emptyset_A$ devono effettivamente simulare i nove che seguirebbero la cifra più significativa letta nel pdc in cui compare $\emptyset_A$ per cui, in questo caso, la presenza di $\emptyset_A$ e di C/N = C consente al complementatore di agire, trasformando gli zeri introdotti a monte in nove che, come tali, giungono al sommatore.

E' da notare il fatto che l'aggiungitore di unità nel canale di accumulatore non interviene mai nell'esecuzione delle operazioni aritmetiche. Nel caso della moltiplicazione può intervenire soltanto nel primo pdc di ogni sottofase durante il quale si esegue il trasferimento delle cifre di moltiplicatore da T_0 a RAM .

3.3.2. L'operazione Moltiplicazione

La fase esecutiva β per l'operazione di moltiplicazione è suddividibile in tante sottofasi quante sono le cifre del moltiplicatore, più una quando l'ultima cifra di moltiplicatore sia $\geqslant 5$ (v. prf. 3.2.5.), che si susseguono ordinatamente senza discontinuità.

Indicheremo con l'indice 1 , 2 , ... , r , (r + 1) , la sottofase in cui sta operando: β_1 , β_2 , ... , β_r , con un secondo indice 1 , 2 , 3 , ... , affiancato al primo, il periodo di cifra in esame: $\beta_{1,1}$; $\beta_{1,2}$; $\beta_{1,3}$ ... $\beta_{2,1}$; $\beta_{2,2}$ ...

Il primo pdc di ciascuna sottofase è destinato al trasferimento della cifra di moltiplicatore dal registro T_0 al Registro Attuale di Moltiplicatore - RAM - costituito da quattro flip-flop che conservano la cifra immagazzinata per tutta la durata della sottofase, dal μs 7 del pdc 1 al μs 0 del pdc 1 della sottofase successiva.

In $\beta_{1,1}$ quindi viene letta in T_0 la cifra di moltiplicatore t_1 che rimane presente all'ingresso del canale di accumulatore da $\beta_{1,1,6}$ a $\beta_{1,2,5}$ ove il terzo indice indica il microsecondo considerato nel pdc indicato del secondo indice.

Se t_1 è $\geqslant$ 5 , cioè se è presente il bit d , viene azionato il

complementatore (a 9) e l'aggiungitore di unità (v. cap.4, prff. 3, 4) in modo che fra il μs 6 ed il μs 7 di $\beta_{1,1}$ la cifra t_1 viene opportunamente trasformata secondo le regole esposte al prf. 3.3.5 . In $\beta_{1,1,7}$ la cifra modificata di moltiplicatore t_1' ($t_1' \leqslant 5$) viene scritta nel RAM .

Se t_1 è $\geqslant 5$ al μs 8,5 di $\beta_{1,1}$ la presenza del bit d all'ingresso del canale di accumulatore consente di far scattare il flip-flop d_{n-1} dalla posizione $\overline{d_{n-1}}$ alla posizione d_{n-1} .

Con la forma d'onda F01 la cifra t_1' è presente all'ingresso 2 del Sommatore mentre all'ingresso 1 vi può essere una configurazione di segnali che combinandosi con t_1' può dare in uscita del sommatore un segnale indesiderato. Occorre quindi impedire che questo segnale si trasferisca in altri organi bloccando con il segnale G in $\beta_{1,1}$ le porte di scrittura negli Staticizzatori di uscita verso accum. e nel contatore dei prodotti.

In $\beta_{1,1}$ si legge anche il primo carattere del numero moltiplicando in memoria i cui segnali sono presenti all'ingresso del canale di Memoria da $\beta_{1,1,6}$ a $\beta_{1,2,5}$, e transitano nel canale stesso, verso il Sommatore, dal $\beta_{1,1,9}$ a $\beta_{1,2,5}$ in accordo alla forma d'onda F02 che apre le porte d'ingresso in caso di moltiplicazione.

In $\beta_{1,1}$ quindi, prima del μs 9 , l'assenza di qualunque segnale in transito sul canale di Memoria significa che è presente uno zero assoluto, il quale può essere mutato nella configurazione della cifra 5 se l'operazione che si sta eseguendo è sottrattiva, quando cioè è azionato il complementatore di Memoria ed il relativo aggiungitore di unità (v. prff. 3.4.1. e 3.4.2.).

Se la cifra di moltiplicatore t_1' è = 5 , nel Sommatore si esegue la somma 5 + 5 = 10 e si origina un riporto decimale che al μs 7 , come sarà detto tra poco, può essere erroneamente immagazzinato nel contatore dei riporti, alterando quindi il risultato dell'operazione in corso. Per questo motivo, quando si è in moltiplicazione, occorre bloccare le uscite del Sommatore nel primo pdc di ogni sottofase.

Il contenuto del RAM viene decodificato da una rete logica a diodi in modo da consentire il transito verso un opportuno organo di rigenerazione degli impulsi M_0 , M_1 , ... M_4 in numero pari alle unità della cifra t_1' immagazzinata in RAM .

Dal rigeneratore escono due treni di impulsi r SP ed UP .

r S.P è formato dagli impulsi $\overline{M}_0$, $\overline{M}_1$, ... $\overline{M}_4$, ossia dagli impulsi M_0 , M_1 , ... M_4 però con polarità invertita, cioè partenti da una base negativa per salire verso lo zero quando l'impulso è presente.

UP è formato dagli impulsi M_0' , M_1' , ... , M_4' ossia dagli stessi impulsi M_0 , M_1 , ... , M_4 non invertiti e ritardati di circa 0,2 µs rispetto ai corrispondenti impulsi di r SP .

Esamineremo ora, sotto forma tabellare, cosa avviene microsecondo per microsecondo nei primi pdc di ogni sottofase.

p.d.c.	µs	
1	8,5	Si trasferisce la cifra zero codice contenuta nel contatore dei Riporti - CR - nello Staticizzatore dei Prodotti - SP -
1	9	Si cancella a zero codice il C.R. e compare la forma d'onda F02 . Con essa la cifra di moltiplicando m_1 , eventualmente modificata transitando attraverso il Complementatore di Memoria ed il relativo aggiungitore di unità, è presente all'ingresso 1 del Sommatore. Attraverso la porta S u b il contenuto di SP è presente all'ingresso 2 del Sommatore e da β 1,1,9 a β 1,2,0 si esegue la somma $m_1 + 0 = m_1$.
2	0	Si cancella a zero assoluto il contenuto di SP per mezzo del primo impulso $\bar{M}_0$ di r SP . Da questo istante cessa di essere corretta l'uscita del sommatore, ma il risultato corretto m_1 della somma appena fatta è conservato per ancora circa 0,4 µs mediante una linea di ritardo inserita all'uscita del sommatore.
2	0,2	Si scrive in SP il risultato della somma eseguita il primo impulso M_0' di UP . I segnali provenienti dal sommatore sono presi dopo la linea di ritardo, quindi sono ancora corretti. Da questo istante fino al µs 1 il contenuto di SP , m_1 , è presente all'ingresso 2 del Sommatore e si somma con m_1 presente all'ingresso 1 : $m_1 + m_1 = 2\ m_1$.
2	1	Si cancella SP con l'impulso $\bar{M}_1$ di r SP .
2	1,2	Si scrive in SP il risultato 2 m_1 con M_1' di UP e, se vi è riporto decimale R , si scrive 1 nel contatore dei Riporti mediante M_1' di UP . Dal µs 1,2 al µs 2 si esegue la somma $m_1 + 2\ m_1 = 3\ m_1$.
2	2	Si cancella SP con $\bar{M}_2$ di r SP .
2	2,2	Si scrive in SP il risultato 3 m_1 mediante M_2' di UP e, con lo stesso M_2' , si aumenta di una unità il contenuto di CR se la somma $m_1 + 2\ m_1$ ha dato luogo ad un riporto decimale.

p.d.c.	µs	
2	3	L'operazione continua ripetendosi tante volte quan-
2	3,2	ti sono gli impulsi UP , ossia t_1' volte.
2	4	Dal µs 4,2 in avanti SP contiene il risultato $t_1'.m_1$
2	4,2	ed il C.R. contiene il riporto complessivo r_1' .
2	5	Cessa di esistere la FO2 e comincia FO1 . Da questo istante il contenuto di SP si presenta, tramite la porta S u a , all'ingresso 1 del Sommatore mentre è precluso l'ingresso dei segnali provenienti da Memoria. Cessa anche di esistere la cifra di moltiplicatore in uscita da accumulatore.
2	6	E' presente in ingresso del canale di accumulatore la cifra a_1 letta in accumulatore in β 1,2,5 , e si presenta all'ingresso 2 del Sommatore in accordo con la FO1 .
2	7	Dal µs 6 al µs 7 , si esegue nel Sommatore la somma $t_1'.m_1 + a_1$ ed al µs 7 questo risultato è scritto negli staticizzatori di uscita verso accumulatore tramite la porta GUA . Al µs 7, l'eventuale riporto decimale viene scritto nel CR il cui contenuto diventa quindi $r_1 = r_1' + 1$.
2	8	Si cancella il contenuto di SP .
2	8,5	Si trasferisce la cifra r_1 contenuta in CR nello SP e ricomincia il ciclo ora descritto.
2	9	Si cancella CR a zero codice. Compare con FO2 la cifra m_2 letta in memoria sul canale di Memoria.
3	0	Si cancella SP
3	0,2	Si scrive in SP il risultato della somma $r_1 + m_2$ Si scrive anche in CR l'eventuale riporto nato da questa somma.
3	1	Si cancella SP .
3	1,2	Si scrive in SP il risultato $(r_1 + m_2) + m_2 =$ $= r_1 + 2\ m_2$, ed in CR l'eventuale nuovo riporto.
3	2	L'operazione prosegue ed in SP si avrà scritto,
3	4,2	al µs 4,2 , il risultato $r_1 + t_1'\ m_2$, mentre nel CR vi sarà la cifra r_2' .
3	5	Il contenuto di SP , $r_1 + t_1'\ m_2$, è presente allo ingresso 1 del Sommatore con la FO1 .

p.d.c.	µs	
3	6	Si presenta all'ingresso 2 del Sommatore la cifra a_2 letta in memoria.
3	7	Si scrive sugli staticizzatori d'uscita il risultato $(r_1 + t_1' m_2) + a_2$, mentre l'eventuale riporto viene scritto in CR il cui contenuto sarà $r_2 = r_2' + 1$
3	8	Si cancella SP
3	8,5	Si trasferisce r_2 in SP etc.

L'operazione descritta continua fino alla comparsa del segnale di fine operazione.

Se il primo carattere proveniente da Memoria in $\beta_{1,1,6}$ era un segno, i risultati parziali e finali ottenuti in $\beta_{1,2,}$, sono privi di significato perciò occorre tenere chiuse le porte che controllano le scrittura negli staticizzatori di uscita e nel contatore dei Riporti.

Il segnale G tiene conto anche di questa eventualità. La cifra di accumulatore a_1 letta in $\beta_{1,2}$ viene, in questo caso, letta di nuovo in $\beta_{1,3}$, per cui il primo risultato utile si avrà in $\beta_{1,3}$ anzichè in $\beta_{1,2}$.

Nel pdc successivo a quello in cui è comparso φ' inizia la sottofase β_2 .

Da $\beta_{2,1,6}$ fino a $\beta_{2,2,5}$ è presente all'ingresso del canale di accumulatore la cifra di moltiplicatore t_2 letta in T_0 .

Dal µs 6 al µs 7 di $\beta_{2,1}$ questa cifra viene modificata eventualmente dal complementatore e dall'aggiungitore di unità di accumulatore, ed al µs 7 viene scritta nel RAM in cui al µs 0 era stata cancellata la precedente cifra t_1' .

Al µs 7,5 il flip-flop d_{n-1} viene riportato nella posizione $\overline{d_{n-1}}$ per essere pronto ad accogliere la nuova indicazione del fatto che t_2 è $\geqslant 5$ in $\beta_{2,1,8,5}$.

Al µs 9 di $\beta_{2,1}$ inizia la F02 e con essa il ciclo di moltiplicazione secondo quanto sopra esposto.

Le sottofasi β_n si susseguono ininterrottamente fino alla com-

parsa del segnale φ di fine generale il quale è determinato dalla presenza contemporanea del segnale φ' di fine operazione, del segnale $\emptyset_R$ di fine moltiplicatore e del segnale $\overline{d_{n-1}}$ del flip-flop d_{n-1}. Se infatti quando compare $\emptyset_R$ il flip-flop è nella posizione d_{n-1} significa che l'ultima cifra di moltiplicatore è $\geqslant 5$ per cui segue una ulteriore sottofase durante la quale la cifra di moltiplicatore è 0 e viene simulata dal generatore di zeri in ingresso del canale di accumulatore. Transitando attraverso l'aggiungitore di unità questo zero diventa un 1 che viene scritto in RAM : la sottofase β_{r+1} segue lo svolgimento normale.

Le regole secondo cui intervengono il complementatore e l'aggiungitore di unità sul canale di accumulatore durante il transito della cifra di moltiplicatore sono le seguenti.

Quando la cifra t_h è $\geqslant 5$ deve essere complementata a 10 e si deve aggiungere una unità alla cifra successiva t_{h+1} (v. prf.3.2.5.).

Siccome la complementazione a 10 è fatta in macchina complementando a 9 ed aggiungendo una unità si vede che:

1) - La complementazione a 9 avviene ogni volta che t_h è $\geqslant 5$.

2) - L'aggiunta di una unità avviene in due casi distinti:
   a) quando $t_h \geqslant 5$ e $t_{h-1} < 5$.
   b) quando $t_h < 5$ e $t_{h-1} \geqslant 5$.

Infatti se $t_h \geqslant 5$ e $t_{h-1} \geqslant 5$ allora occorrerebbe fare

$$10 - (t_h + 1) = 9 - t_h$$

e quindi non occorre aggiungere 1 .

3.3.3. L'operazione di trasferimento

Si hanno due tipi di trasferimento:

1) Trasferimento da Memoria ad Accumulatore od a T (TMA ; TMT).

2) Trasferimento da Accumulatore a Memoria (TAM).

Il trasferimento da Memoria verso accumulatore si esegue tramite un canale controllato dalla porta GMA che sorpassa tutti gli orga

ni di calcolo e permette di scrivere le cifre in arrivo negli staticizzatori di uscita verso accumulatore.

I segnali in transito per questo canale non subiscono nessuna modifica.

Durante la fase β di esecuzione in ogni pdc si legge la cifra in Memoria al μs 5 e dal μs 6 in avanti è disponibile per essere utilizzata a valle. Al μs 7 si apre la porta GMA e si scrive la cifra letta negli staticizzatori di uscita.

Durante le operazioni di trasferimento rimane chiusa la porta che controlla i segnali in uscita del Sommatore.

Il trasferimento da Accumulatore verso Memoria avviene tramite il normale canale di accumulatore attraverso cui i segnali presenti in ingresso dal μs 6 al μs 5 del pdc successivo vengono fatti passare in accordo alla forma d'onda FO1 e giungono, oltre che allo ingresso 2 del Sommatore, agli staticizzatori di uscita verso memoria entro i quali la scrittura è controllata dalla porta GAM .

Questi trasferimenti avvengono soltanto in fase β ; gli staticizzatori di uscita verso memoria accolgono la cifra in arrivo da accumulatore al μs 7 .

Quando il numero in Accumulatore è provvisto di segno (v.prf. 3. 4.3.3.) durante il trasferimento da accumulatore a memoria tale segno viene trasferito in β_1 (l'indice indica il pdc della fase).

Per far questo vi è un apposito generatore di segno che immette i suoi segnali sul canale di accumulatore, a valle dell'aggiungitore di unità, e provvede a fornire sulle linee dei bit a - b - c - d gli impulsi 1 0 1 0 se il segno del numero in accumulatore è positivo, e gli impulsi 1 0 1 1 se il segno è negativo. Questo organo agisce soltanto nel primo pdc se il segno è presente.

Quando il numero in accumulatore è espresso in complemento occorre, durante il trasferimento, complementarlo azionando il complementatore a 9 e l'aggiungitore di unità per trasformarlo in valore assoluto, soltanto nel caso in cui si sia operato su numeri segnati, soltanto cioè quando esista contemporaneamente il segno di Accumulatore.

L'aggiungitore di unità agisce soltanto per tutte le cifre che precedono la prima cifra significativa diversa da zero, questa compresa, secondo quanto esposto al prf. 3.2.3.
Se la Norma contenuta nell'accumulatore indica il complemento del risultato ottenuto nelle precedenti operazioni, ma nessuno dei numeri precedentemente trattati era provvisto di segno, essa viene trasferita in Memoria inalterata, senza complementazione.

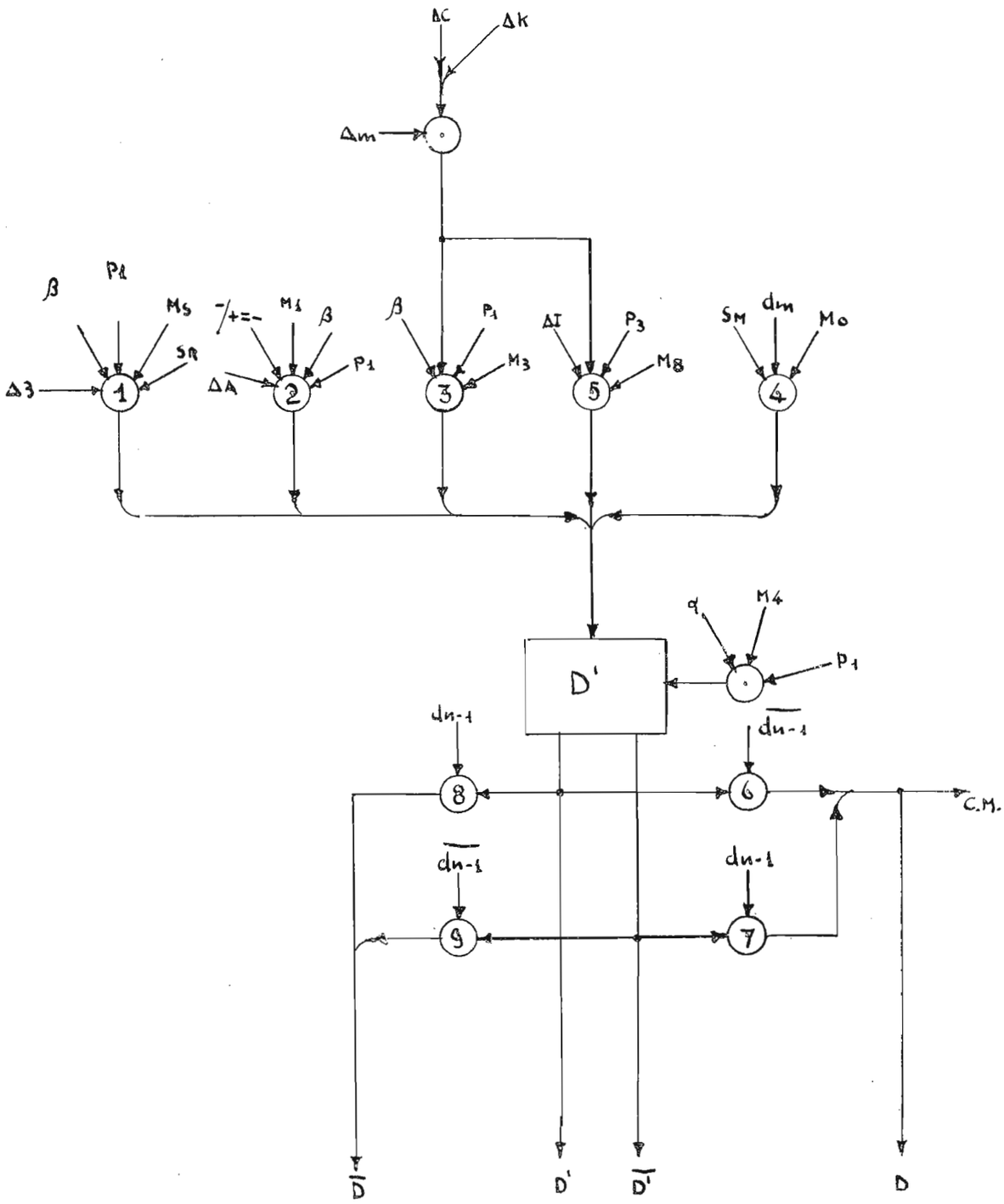

ΔC
Δk
Δm
β
P1
M5
S1
Δ3
1
-/+=-
M1
β
ΔA
2
P1
β
P1
M3
3
ΔI
P3
M8
5
SM
dm
Mo
4
q
M4
P1
D'
dn-1
8
dn-1
6
C.M.
dn-1
9
dn-1
7
D
D'
D'
D

Fig. 3.2

3.4. Descrizione degli organi e del loro funzionamento con riferimento alle circostanze che lo determinano.

3.4.1. Ordine di Complementazione "a 9" per la cifra proveniente da Memoria.

Il segnale D , la cui presenza consente la complementazione, è presente ogni volta che la cifra proveniente da memoria deve essere sottratta dalla corrispondente cifra letta in accumulatore.

Le circostanze che determinano la presenza di D sono distinguibili in due gruppi, a cui corrispondono due parti distinte del circuito elettronico.

3.4.1.1. I Segni (v. prf. 3.1.2.)

La cifra proveniente da memoria deve essere sottratta dalla corrispondente in accumulatore quando vi è un numero dispari di segni negativi:

$$S_M - \ , \quad S_R - \ , \quad +/- = - \ , \quad S_{op} -$$

L'organo che verifica e memorizza questa condizione è un flip-flop, connesso a contatore , D' , che, dalla posizione di reset $\bar{D}'$ in cui viene riportato all'inizio di ogni operazione in $\alpha\, p_1 M_4$ e che indica assenza di complementazione, commuta la propria posizione ogni volta che gli arriva un impulso dalle porte 1 ... 5 (v. fig.3.2.).

Soltanto se questi impulsi, ciascuno indicante la presenza di un segno negativo fra quelli elencati, sono in numero dispari il f.f. rimane nella posizione D' che consente la complementazione

La porta 1 permette il passaggio dell'impulso M5 nella fase preparatoria α di moltiplicazione - Δ 3 - quando il segno di moltiplicatore è negativo S_R- .

La porta 2 permette il passaggio dell'impulso M_1 quando il segno di accumulatore è negativo $+/- = -$.

La porta 3 permette il passaggio dell'impulso M_3 quando il segno dell'operazione è negativo $\Delta_m = S_{op} -$.

La porta 4 permette il passaggio dell'impulso M_0 quando il segno del numero in memoria è neg. $S_M \cdot d_m = S_M -$.

La porta 5 permette il passaggio dell'impulso M_8 quando si esegue una istruzione IT sottrattiva.

La temporizzazione degli impulsi è quella risultante dalla figura; l'intervallo di tempo per due impulsi successivi è determinato dalle caratteristiche elettroniche del flip-flop .

Il segnale D' o $\bar{D}$' è presente per tutta la durata di una operazione.

3.4.1.2. In moltiplicazione (v. prf. 3.2.6.)

Quando la cifra di moltiplicatore letta nel registro T_0 è $\geqslant 5$ occorre invertire la decisione presa dal flip-flop D' , trattando questa nuova condizione nello stesso modo in cui sono stati trattati i segni.

A differenza di quanto visto per i segni però, la decisione assunta in funzione di questa nuova condizione deve durare per tutta la sottofase β_h in cui si opera la cifra di moltiplicatore m_h in esame, senza interessare le altre sottofasi per le quali invece è sempre valida la decisione presa dal flip flop D' .

L'indicazione del fatto che $m_h \geqslant 5$ è contenuta nel flip flop d_{n-1} le cui uscite, combinate con quelle del f.f. D' mediante le porte 6 , 7 , 8 , 9 , determinano la presenza del segnale D o del segnale $\bar{D}$ in accordo con la tabella seguente.

FF d_{n-1} \ FF D'	D'	$\bar{D}$'
d_{n-1}	$\bar{D}$	D
$\overline{d_{n-1}}$	D	$\bar{D}$

3.4.2. Ordine "aggiungi una unità" alla cifra proveniente da memoria.

Il segnale +1 che consente la somma di una unità alla cifra in transito sul canale di Memoria dipende da due circostanze.

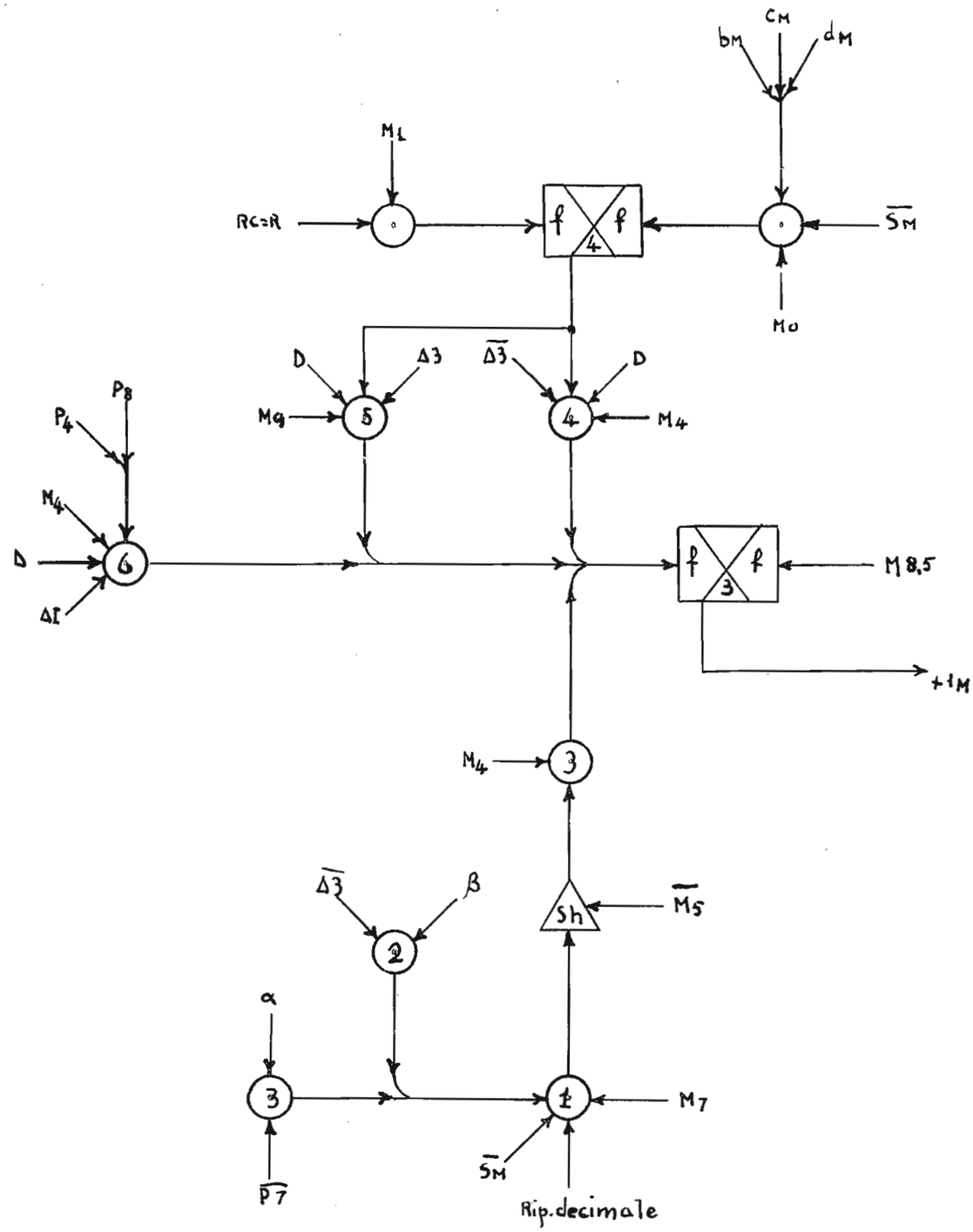

cM
bM
dM
M1
RC=R
f
f
4
$\overline{S_M}$
M0
D
Δ3
$\overline{\Delta 3}$
D
P8
P4
M9
5
4
M4
M4
D
6
ΔI
f
f
3
M8,5
+1M
M4
3
$\overline{\Delta 3}$
β
Sh
$\overline{M_5}$
2
α
3
1
M7
$\overline{P_7}$
$\overline{S_M}$
Rip.decimale

Fig. 3.3

3.4.2.1. Riporto decimale di somma (v. prf. 3.2.1.)

Quando, durante una operazione di somma, dal Sommatore si ottiene un riporto di decina, occorre sommare una unità alla cifra proveniente da Memoria.

Il segnale di riporto R tramite la porta 1 (v.fig.3.3.) viene scritto e conservato nello staticizzatore S_h dal µs 7 fino al µs 5 del pdc successivo; la porta 3 consente all'impulso M_4 di far scattare il f.f. 3 nella posizione +1 , a cui corrisponde il consenso all'aggiunta di una unità, quando lo staticizzatore S_h lo permetta. La porta 1 è controllata dai segnali provenienti dalle porte 2 e 3; questi dicono che la porta 1 è sempre aperta in fase esecutiva β purchè non si stia eseguendo una moltiplicazione (porta 2) nel quale caso tutti i riporti di decina confluiscono nel contatore dei riporti decimali. Durante la fase preparatoria α la porta 1 è sempre aperta, tranne che durante il pdc settimo, in cui si legge l'ultimo carattere dell'indirizzo della parola in memoria. (porta 3).

3.4.2.2. Complementazione a 10

Quando, per una delle circostanze elencate nel cap.3.4., è presente il segnale D che ordina la complementazione "a 9" della cifra proveniente da Memoria occorre aggiungere una unità alla cifra in transito, per tutte le cifre, fino a che compare la prima cifra significativa diversa da zero, questa compresa (vedi nota del prf. 3.2.2.).

L'impulso di comando arriva al f.f. 3 tramite la porta 4 quando non si è in moltiplicazione e attraverso la porta 5 quando si sta eseguendo una operazione di moltiplicazione.

Il f.f. 4 è riportato nella posizione in cui consente il passaggio dall'impulso di comando attraverso le porte 4 e 5 alla fine di ogni operazione, quando compare il segnale RC = R , e commuta la propria posizione quando arriva dalla memoria la prima cifra significativa diversa da zero, che non sia il carattere del segno. Per la temporizzazione scelta la commutazione è efficace a partire dal pdc successivo a quello in cui è comparsa la condizione necessaria, in modo che anche per la prima cifra significativa diversa da zero è operante il comando +1 .

La porta 6 consente l'aggiunta di una unità al pdc 8 quando si stiano eseguendo istruzioni del tipo IT ; le quali si svolgono

soltanto dal pdc 4 al pdc 8 della fase α . Se la prima cifra in transito al pdc p_4 durante questa operazione è uno zero, l'aggiunta di una unità nel periodo successivo è ottenuta mediante il normale canale dei riporti, secondo quanto esposto più avanti. Soltanto nel pdc 8 , essendo bloccato il canale del riporto nel pdc 7 , occorre provvedere al comando +1 mediante la porta 6 .

3.4.2.3. Temporizzazioni

Il tempo utile per la modifica della cifra in transito attraverso l'aggiungitore di unità va dal µs 6 al µs 7 di ciascun pdc in tutte le operazioni fuorchè in quella di moltiplicazione. Pertanto, se la cifra deve essere complementata a 10, occorre che il comando +1 sia operante prima del µs 6, momento in cui la cifra è disponibile all'ingresso del canale di memoria.

Attraverso la porta 4 l'impulso del set del f.f. 3 passa al µs 4. Al µs 8,5 il f.f. 3 viene riportato nella posizione +1 mediante l'impulso M8,5 per essere pronto ad accogliere, al µs 4 del pdc successivo l'indicazione derivante dall'eventuale riporto decimale conservato nello staticizzatore S_h .

Fintanto che le cifre provenienti dalla memoria sono gli zeri il f.f. 4 permane nella posizione di set e l'impulso M4 passa ad ogni pdc . Però, per effetto della complementazione a 10 lo zero viene trasformato in 9 attraversando il complementatore e poi viene di nuovo trasformato in zero attraverso l'aggiungitore di unità, generando contemporaneamente un riporto decimale che stabilisce nuovamente l'ordine +1 per il pdc successivo. In questo caso, l'impulso M4 proveniente dalla porta 4 sarebbe superfluo.

Quando dalla memoria arriva al µs 6 la prima cifra significativa diversa da zero il f.f. 4 mantiene inalterata la propria uscita fino al µs 0 , M_0 , del successivo pdc in modo che per la cifra in transito si effettua l'aggiunta di una unità, ma non per le successive, a meno che questa aggiunta non dipenda dai riporti decimali.

In moltiplicazione l'aggiunta di una unità per effetto di complementazione a dieci deve eseguirsi dal µs 9 al µs 5 del pdc successivo, per tutto il tempo cioè in cui, con la forma d'onda F02 , la cifra da Memoria è presente all'ingresso 1 del sommatore.

Pertanto il segnale di comando del f.f. 3 arriva dalla porta 5 al µs 9 poichè il segnale D di consenso alla porta 5 è pronto

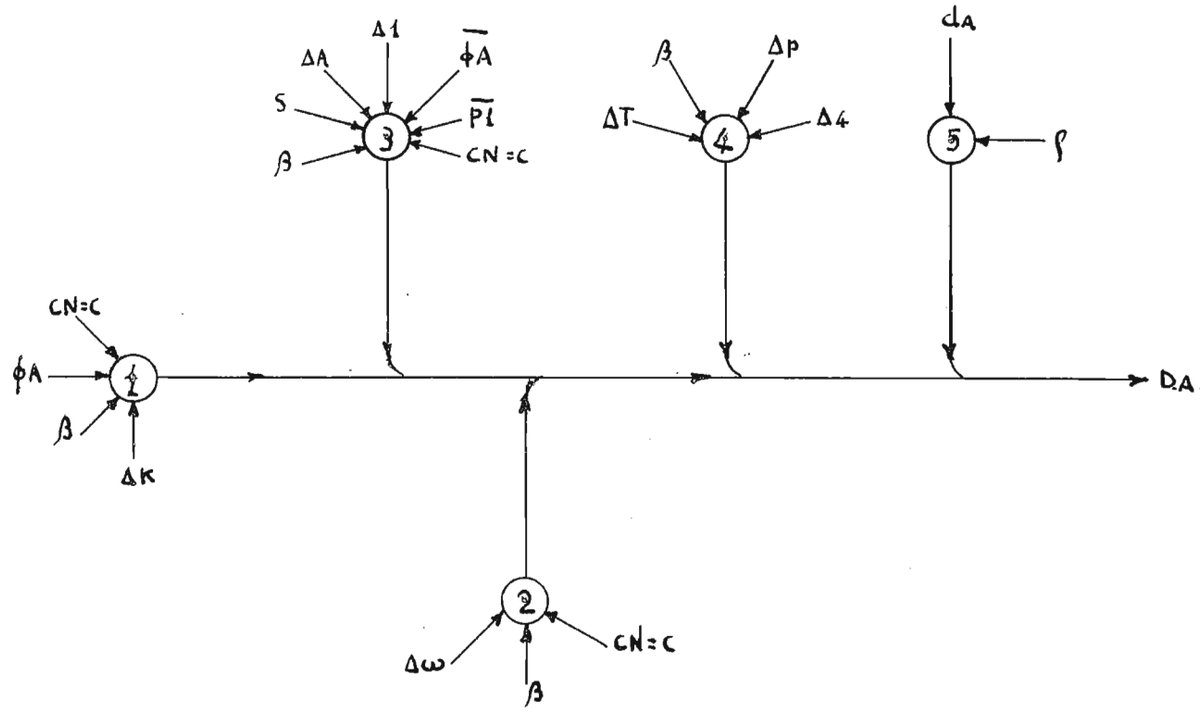

Δ1
ΔA
$\overline{\phi}$A
S
$\overline{P1}$
β
CN=C
3
β
Δp
ΔT
Δ4
4
dA
5
ρ
CN=C
φA
1
β
ΔK
DA.
2
Δω
CN=C
β

Fig. 3.4

soltanto al μs 8,5 quando, nel primo pdc di ciascuna sottofase βn , il f.f. d_{n-1} commuta la propria posizione per effetto del cambiamento della cifra di moltiplicatore.

3.4.3. Ordine di complementazione per la cifra letta in Accumulatore.

Il segnale D_A che consente la complementazione "a 9" della cifra in transito proveniente dall'accumulatore appare quando si verificano le circostanze seguenti. (V. fig. 3.4.).

1) - Fine di accumulatore durante una operazione aritmetica (ΔK)

Quando una operazione aritmetica si protrae oltre il pdc in cui compare il segnale $\emptyset_A$ di fine accumulatore e vi è indicazione (CN = C) del fatto che il numero in accumulatore è in complemento, per cui gli zeri introdotti da apposito generatore in nove. Porta 1 .

2) - Esecuzione dell'istruzione FAM ($\Delta \omega$)

Durante l'esecuzione dell'istruzione FAM occorre poter considerare le cifre provenienti dall'accumulatore nel loro valore normale, cosicchè, se vi è indicazione CN = C , occorre complementare. Porta 2 .

3) - Trasferimenti da Accumulatore a Memoria (Δ A . Δ1)

Il numero contenuto in accumulatore sotto forma di complemento (CN = C) viene trasferito in memoria senza essere riportato in forma normale soltanto quando non vi è indicazione di segno dell'accumulatore (S). In questo caso si opera sui complementi come se fossero numeri normali positivi.

Quando invece l'accumulatore è segnato occorre complementare il numero in transito per riportarlo in forma normale. La complementazione non viene eseguita nel primo pdc ($\bar{p}_1$) perchè in tale pdc si trasferisce il carattere del segno, e nessuna informazione arriva dall'accumulatore essendone bloccato il pilotaggio. Il segno viene introdotto nel canale di accum. da apposito generatore (V. prf. 3.4.1.1.).

La complementazione cessa quando compare il segnale $\emptyset_A$ di fine parola in accumulatore perchè le cifre successive sono degli zeri introdotte dal generatore di zeri, e tali devono rimanere. Porta 3 .

4) - Esecuzione dell'istruzione ITT (Δ 4.Δp.ΔT)

L'istruzione ITT prevede che il numero proveniente da un registro T , T_b , quindi transitante per il canale di accumulatore, venga sottratto ad un numero (I mod. T_a) proveniente dal canale di memoria, invertendo quindi le funzioni di sottraendo e minuendo che normalmente assumono i numeri provenienti da accumulatore e da memoria (v. prf. 3.2.3.).

Pertanto occorre complementare il sottraendo, cioè il numero proveniente dai registri T , mediante la porta 4 .

5) - In moltiplicazione (v. prf. 3.2.5.)

Nel primo pdc di ogni sottofase β_h di esecuzione della moltiplicazione si trasferisce la cifra di moltiplicatore m_h dal registro T_o al registro RAM . Se essa è $\geqslant 5$, ossia se è presente il bit d fra i segnali provenienti dal registro T_o , occorre complementarla, per cui l'ordine D_A trae origine dalla porta 5.

3.4.4. Comando +1 sul canale di accumulatore

3.4.4.1. L'ordine +1 che consente l'aggiunta di una unità alla cifra in transito sul canale di accumulatore è azionato nelle seguenti circostanze:

a) Ogni volta che la cifra proveniente dell'accumulatore è complementata a 10.
b) Quando occorre aumentare di una unità la cifra di moltiplicatore in transito (v. prf. 3.3.2.).

3.4.4.2. Circostanza a)

Questa circostanza prevede tre casi:

a_1) Trasferimenti da Acc. a Mem. di numeri espressi in complemento, CN = C , in presenza di segno. Porta 1 .

a_2) Istruzioni del tipo ITT (ΔT - Δ p). Porta 2 .

a_3) Istruzione FAM ($\Delta\omega$) quando CN = C . Porta 3 .

Le condizioni che debbono essere verificate affinchè in queste circostanze l'ordine +1 sia presente sono la permanenza del segnale = 0 del f.f. 1 e la fase β .

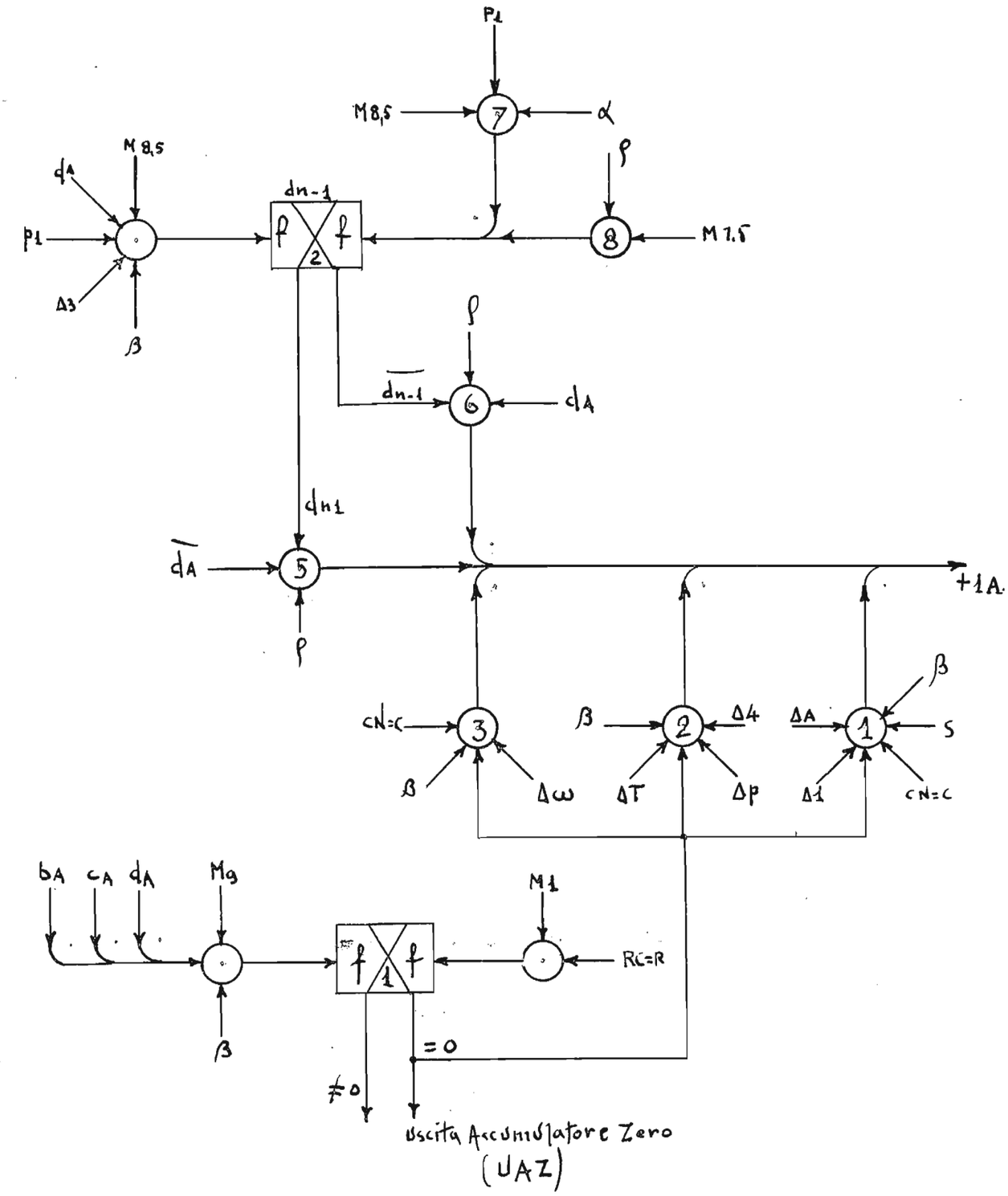

P1
M8,5
α
ρ
M7,5
dA
M8,5
dn-1
P1
Δ3
β
ρ
dn-1
dA
dn1
dA
+1A
ρ
β
CN=C
β
Δ4
ΔA
S
β
Δω
ΔT
Δp
Δ1
CN=C
bA
cA
dA
Mg
M1
RC=R
β
=0
≠0
Uscita Accumulatore Zero
(UAZ)

Fig. 3.5

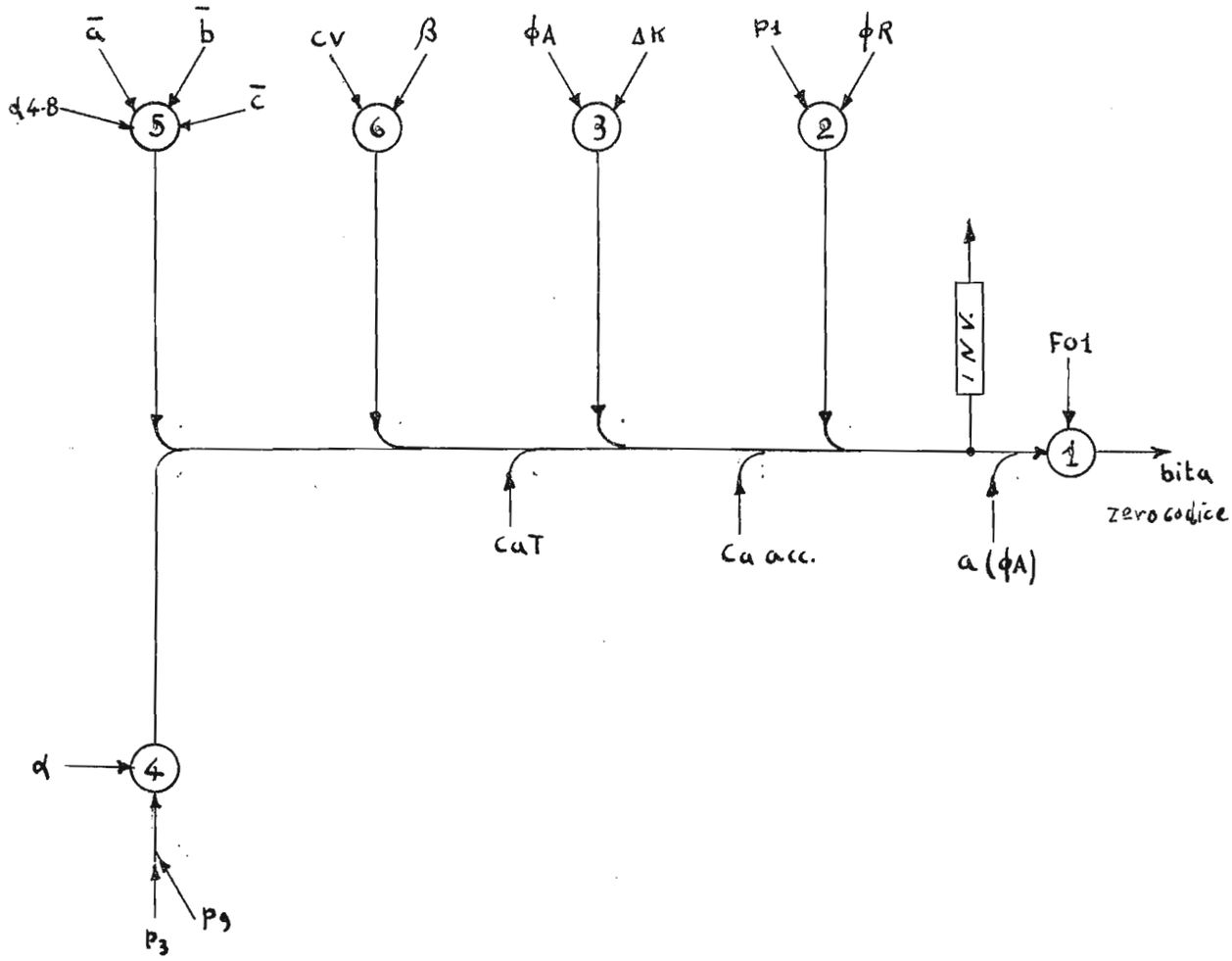

ā
b̄
c̄
ϕ4-8
5
CV
β
6
ϕA
ΔK
3
P1
ϕR
2
I N V.
Fo1
1
bita
zero codice
CaT
Ca acc.
a (ϕA)
α
4
P3
P9

Fig. 3.6

Il f.f. 1 commuta la propria posizione quando dall'acc. giunge la cifra meno significativa $\neq 0$ del numero operando. La temporizzazione è scelta in modo che gli effetti della commutazione siano operanti soltanto per le cifre successive alla prima $\neq 0$ (M9) (V. fig. 3.5.).

Il f.f. 1 viene riportato nella posizione di reset, ossia sul lato = 0 , alla fine di ogni fase, quando compare il segnale RC = R .

3.4.4.3. Circostanza b)

Il f.f. 2 conserva l'indicazione del fatto che la cifra di moltiplicatore precedente quella in transito era $\geqslant 5$ (presenza del d_A); il consenso alla comparsa del segnale $+1_A$ è dato dalle porte 4 e 5 che agiscono soltanto nel primo pdc di ciascuna sottofase di moltiplicazione ($\rho = \beta \cdot p_1$).

La combinazione dei segnali d_A e d_{n-1} traduce le regole già esposte al prf. 3.3.2. .

Il f.f. 2 viene riportato nella posizione di reset all'inizio della fase α , porta 7 , ed al μs 7,5 di ogni $\beta \cdot p_1$, porta 8, dopo cioè che la cifra in transito è stata immagazzinata in RAM ma prima che si perda la disponibilità d'ingresso del can. di acc. della cifra di moltiplicatore attuale.

La presenza del bit d_A in questa cifra consente che il f.f. 2 venga riportato dalla posizione $\overline{d_{n-1}}$ a quella d_{n-1} al μs 8,5 , porta 9 .

3.4.5. Generatore di zeri in uscita dell'accumulatore

Agisce consentendo, tramite la porta 1 , la presenza del segnale F01 sul canale "a" (in modo da realizzare la configurazione "zero codice"), quando sono presenti le seguenti circostanze (fig. 3.6.).

1 - In moltiplicazione quando, dopo la comparsa del segnale Φ_R segue una ulteriore fase ρ . Porta 2 .

2 - Durante le operazioni additive e sottrattive (+MA, +X, -MA, -X, +MT, -MT, +IT, -IT) caratterizzate dalla presenza del segnale Δ K , quando, dopo la comparsa del segnale Φ_A le

operazioni debbono proseguire: Porta 3 .

3 - Durante α 3 e α 9, per permettere il transito attraverso l'U.A. dell'indirizzo del T modificatore e di Ld . Porta 4.

4 - Durante α 4,8 - periodo di modificazione dell'istruzione - se non avviene modifica (o per scelta di T inesistente, o perchè T contiene uno zero assoluto): Porta 5 .

5 - Durante β, se è presente il segnale CV - condizione di salto verificata - per consentire il transito attraverso l'U.A. del contenuto dei registri W . Porta 6 .

4. G O V E R N O

4.1. Premessa

Il Governo della Macchina si articola in tre gruppi di organi funzionalmente separati:

- forme d'onda
- registri
- smistamento logico

Il gruppo di organi di forme d'onda ha il compito di generare i segnali temporizzatori che cadenzano il funzionamento di tutti gli organi della macchina.

I registri sono destinati ad immagazzinare ed elaborare gli indirizzi da presentare agli organi di selezione di Memoria e di Accumulatore. Esistono inoltre:

- il Registro di Lunghezza, destinato ad immagazzinare, durante la fase di preparazione, il numero indicante la lunghezza dell'operando.
- il Registro di Funzione, destinato ad immagazzinare, durante la fase di preparazione, il codice dell'istruzione che si sta preparando.

Gli organi di smistamento logico devono preparare i comandi che attivano tutti gli organi della macchina, elaborando, attraverso flip-flop e gruppi di porte And e Or, i segnali provenienti essenzialmente dagli altri organi di Governo.

4.2. Forme d'onda

4.2.1. Generalità

I segnali temporizzatori sono di tre tipi, a seconda della durata di tempo che individuano:

- segnali di fase, che durano alcuni p.d.c.
- segnali di p.d.c., che durano un p.d.c. (10 µs per le normali operazioni di macchina) (*).
- segnali mastri, che durano una piccola frazione di p.d.c. (0,4 µs).

Tutti questi segnali sono negativi se presenti. I mastri hanno forma triangolare con durata alla base di circa 0,4 µs . Gli altri segnali sono trapezoidali con fronti di discesa e salita minori di 0,5 µs . L'inizio e la fine di tali segnali trapezoidali si riferiscono sempre all'istante in cui il livello del segnale comincia a variare.

I segnali mastri sono sempre presenti a macchina accesa.

4.2.2. I Mastri (disegno n° 4.1.)

La cadenza fondamentale di macchina è stabilita da un oscillatore principale a 100 kHz, che invia impulsi a distanza di 10 µs uno dall'altro su una linea di ritardo di 10 µs . Dalle prese intermedie di tale linea si prelevano i mastri a 0,5 µs di distanza l'uno dall'altro.

I mastri sono contrassegnati dalla lettera M seguita dal numero del µs che individuano entro il p.d.c. con il loro primo fronte: si hanno perciò i Mastri M 0, M 0,5 M 1,5 etc.

4.2.3. Gli impulsi di p.d.c. (disegno n° 4.2.)

Il mastro M 0 è inviato ad un divisore per 100, detto contatore di p.d.c. Le prime dieci configurazioni delle cento possibili di tale contatore sono decodificate e costituiscono i segnali di p.d.c., contrassegnati dalla lettera p seguita dal numero d'ordine del p.d.c. individuato.

(*) - Le operazioni che involgono l'uso di organi di ingresso-uscita hanno p.d.c. di diversa durata.-

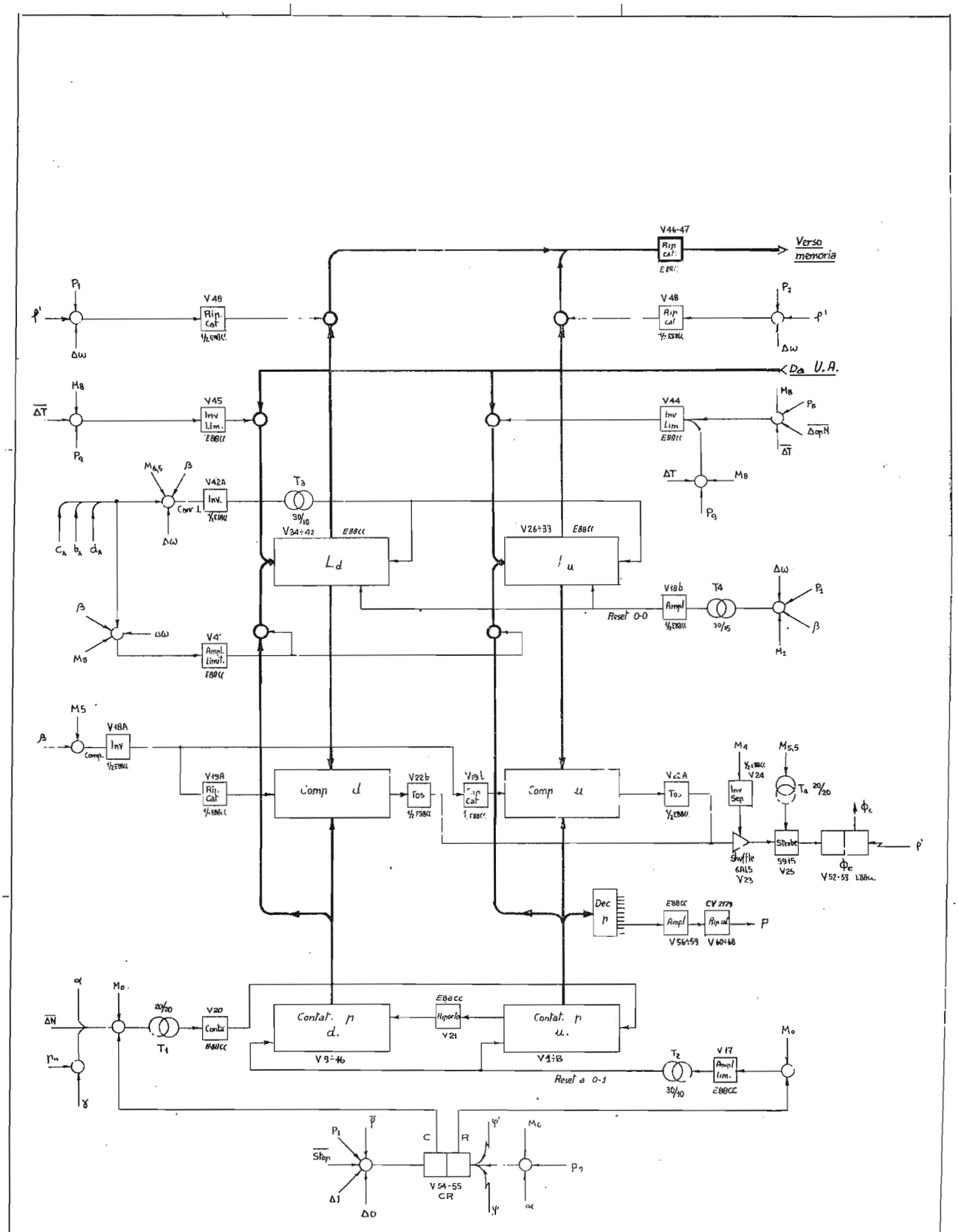
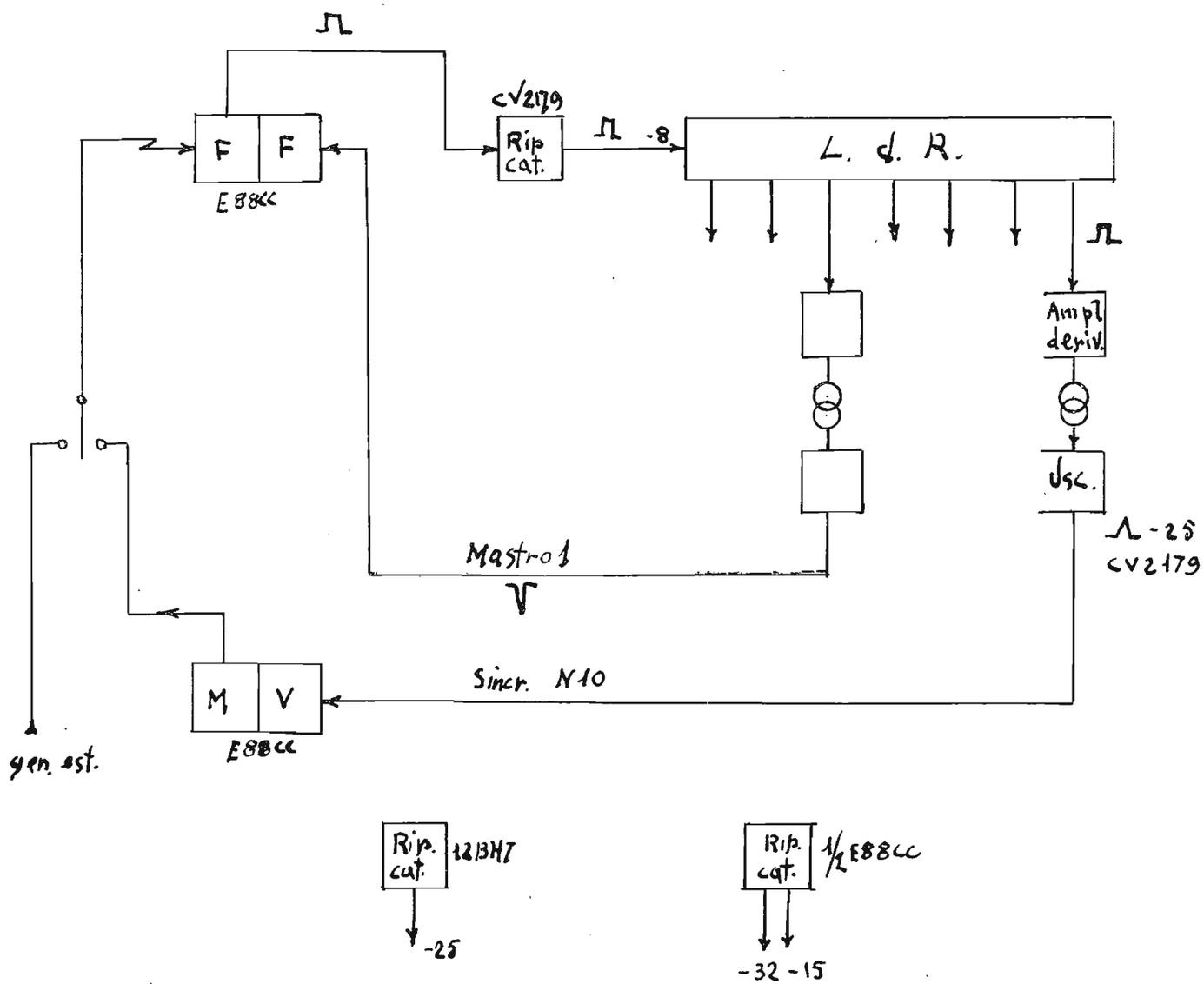

Verso memoria
Da U.A.
V46-47
V48
V45
V44
V42A
V34÷42
V26÷33
Ld
Iu
V18b
T4
Reset 0-0
V18A
V19A
Comp d
V22b
V19b
Comp u
V22A
V24
V23
V25
V52-53
Dec p
V56-59
V60-68
V20
T1
V9÷16
V1÷8
V21
T2
V17
Reset a 0-1
V54-55
CR

Fig. 4.2



GENERATORE DI MASTRI MII - Shema a blocchi

Fig. 4.1

Si conviene che la posizione iniziale del contatore sia 01 e la finale 00 : gli impulsi di p.d.c. vanno pertanto da p1 a p9 .

Ogni impulso di p.d.c. inizia al µs 0 e termina al µs 10 .

Il contatore di p.d.c. può eseguire i seguenti comandi:

- Rim._a_1_: si porta alla configurazione 01, ad M0, qualunque fosse la configurazione contenuta.
- Conta p : conta una unità avanti nel codice di macchina, ad M0 .
- p — RL_: copia in parallelo il proprio contenuto sul registro di lunghezza, (previamente cancellato), ad M8 .

Il contatore di p.d.c. presenta infine permanentemente il proprio contenuto ad uno degli ingressi del comparatore di lunghezza.

4.2.4. I segnali di fase

I segnali di fase comprendono, come accennato, diversi p.d.c. Sono generati da una serie di flip-flop, attivati da opportune reti logiche. Ogni segnale di fase si inizia al µs 8,5 del p.d.c. precedente al gruppo di p.d.c. da individuare, e termina al µs 8,5 dell'ultimo p.d.c. del gruppo da individuare.

Si hanno i seguenti segnali di fase, generati da altrettanti flip-flop:

- segnale di preparazione (o di fase alfa): ha durata fissa e comprende i p.d.c. da p1 a p9 in cui si legge da Memoria una istruzione e la si smista nei vari organi di Governo.
- segnale di esecuzione (o di fase beta): ha durata variabile e comprende i p.d.c. in cui l'istruzione viene eseguita; esso dura pertanto dal periodo 1 al p.d.c. in cui compare il segnale φ'.

4.3. Registri di Governo

4.3.1. Registri di Memoria

I registri di Memoria hanno tutti capacità di quattro cifre decimali (unità, decine, centinaia, migliaia) dato che l'indirizzo di Memoria è un numero di quattro cifre decimali.

I registri di memoria comprendono un Registro Attuale e due Registri Ausialiari. Il Registro Attuale, contrassegnato dal simbolo W, presenta agli organi di comando Memoria l'indirizzo della cella da chiamare. Il primo registro ausiliario, I, conserva, nella fase di esecuzione, l'indirizzo dell'istruzione che si sta eseguendo, per poter passare all'istruzione successiva. Il secondo registro ausiliario, O, conserva, durante la fase di preparazione, l'indirizzo dell'operando da memoria (indicato nell'istruzione).

Il registro W può eseguire i seguenti comandi:

Canc. W : è cancellato ad M0 o eccezionalmente ad M5

Int. Raus → W : riceve in parallelo il contenuto di uno dei due registri ausiliari O od I ad M 1,5 (previa cancellazione).

+ 1 W : conta una unità avanti nel codice della macchina, ad M 3 .

- 1 W : conta una unità indietro nel codice della macchina ad M3

+ 10 : conta una decina avanti, sempre ad M3

Scrit. W → Raus : copia il proprio contenuto in parallelo su uno dei registri ausiliari da M7 ad M10, senza disturbare la selezione di Memoria (che avviene da M4,5 ad M 10).

Usc. W → U.A. : emette il proprio contenuto serialmente (una cifra per p.d.c.) verso U.A.

Per le operazioni che interessano anche i registri ausiliari, oltre al comando di operazione, si richiede un comando di selezione di registro, sel 0 o sel 1. Tale comando agisce su un flip flop selettore di registro: pertanto ogni registro, una volta selezionato, resta in tale condizione fintantochè non venga selezionato l'altro. Il mastro di selezione varia a seconda delle operazioni. Il registro W presenta con continuità il proprio contenuto alla Memoria.

Ognuno dei registri ausiliari può eseguire i seguenti comandi:

Canc. Raus M : è cancellato da M 1,5 a M 4,5

Int. Raus → W : trasferisce il proprio contenuto in parallelo su W , restando cancellato.

Scritt. W → Raus : copia in parallelo il contenuto di W (previa cancellazione) da M7 ad M10 .

Scritt. U.A. → Raus M : scrive serialmente (una cifra decimale per p.d.c.) l'uscita da U.A. da M7 a M10 (previa cancellazione).

Tutti questi comandi vanno associati naturalmente ad un comando di selezione di registro.

4.3.2. Registri di Accumulatore

I Registri hanno tutti capacità di due cifre decimali (unità e decine), dato che l'indirizzo di Accumulatore è un numero di due cifre decimali. (♠).

Anche i Registri di Accumulatore comprendono un Registro Attuale e due Registri Ausiliari.
Il Registro Attuale, contrassegnato dal simbolo V, presenta agli organi di comando Accumulatore l'indirizzo della cella da chiamare.
I due registri ausiliari, P ed R, memorizzano indirizzi utili per alcune istruzioni particolari (moltiplicazione, etc.).
Il registro V può eseguire i seguenti comandi:

Canc. V : è cancellato ad MO .

Int. Raus → V : riceve in parallelo il contenuto di uno dei due registri ausiliari P o R, ad M 1,5 (previa cancellazione).

+ 1 V : conta una unità avanti nel codice di macchina ad M 3 .

Scritt. V → Raus : copia il proprio contenuto in parallelo su uno dei due registri ausiliari da M7 a M10, senza disturbare la selezione di Accumulatore, che avviene da M4,5 a M10 .

Zero dec. : scrive zero codice sulle decine di V (previamente cancellate) ad M 1,5 .

(♠) - La selezione fra Accumulatore e Modificatore è affidata ad un comando separato (vedi 2.2.3.).

$V_A \rightarrow V_d$: scrive l'uscita da U.A. sulle decine di V (previamente cancellate) ad M3 .

$d \rightarrow V_u$: scrive UNO binario nella posizione d delle unità ad MO (previa cancellazione).

Per le operazioni che interessano anche i registri ausiliari, oltre al comando di operazione si richiede un comando di selezione di registro: sel P o sel R . Tale comando agisce su un flip flop selettore di registro: pertanto ogni registro, una volta selezionato, resta in tale condizione fintantochè non venga selezionato l'altro. Il mastro di selezione è diverso a seconda delle operazioni.
Il Registro V presenta con continuità il proprio contenuto all'Accumulatore.
Ognuno dei registri ausiliari può eseguire i seguenti comandi:

Int. Raus → V : trasferisce il proprio contenuto in parallelo su V a M 1,5 , restando cancellato.

Scritt. V → Raus A : copia in parallelo il contenuto di V (previa cancellazione) da M7 a M10.

Scritt. U.A. → Raus A : scrive serialmente (una cifra ad ogni p.d.c.) l'uscita da U.A. da M7 a M10 (previa interrogazione).

Zero Raus A : scrive in parallelo zero codice sulle unità e sulle decine ad M5 (previa interrogazione).

Tutti questi comandi vanno naturalmente associati al comando di selezione di registro.

4.3.3. Il Registro di Lunghezza (disegno n° 4.2)

Ha la capacità di due cifre decimali (unità e decine).
Le sue funzioni principali sono:

a) - ricevere il numero di due cifre indicante la lunghezza dell'operando all'atto della preparazione.
b) - presentare questo numero ad ogni p.d.c. dell'esecuzione, ad uno degli ingressi del Comparatore di Lunghezza.

All'altro ingresso del Comparatore di Lunghezza giunge con continuità il contenuto del Contatore di p.d.c. (vedi 4.2.3.): al M5,5 del p.d.c. in cui si ha coincidenza fra i due ingressi, il Comparatore emette un segnale di fine dell'operando .
Il Registro di Lunghezza può ricevere informazioni in parallelo

dal contatore di p.d.c. e può emettere in serie (una cifra per p.d.c.) verso la Memoria, tramite U.A.

In complesso il Registro di Lunghezza può eseguire i seguenti comandi:

Canc. RL : è cancellata ad M2, oppure ad M 6,5

UA → Lu : si riempie in serie ad M8 da U.A. (previa cancellazione).

Zero RL : scrive zero codice in parallelo sulle unità e decine ad M2 (previa cancellazione).

Conta p → RL : copia in parallelo il contenuto del Contatore p ad M8 .

RL → UA : emette il proprio contenuto in serie su U.A.

Come già detto, il Registro di Lunghezza presenta con continuità il proprio contenuto ad uno degli ingressi del Comparatore di Lunghezza.

4.3.4. Il Registro di Funzione

Ha capacità di due cifre decimali (unità e decine).

Il suo compito è di ricevere da Memoria il codice di istruzione durante la preparazione e di trasmetterlo in continuazione, opportunamente decodificato, a tutti gli organi della macchina.
Il Registro di Funzione può eseguire i seguenti comandi:

Canc. RF : è cancellato ad M5

Scritt. da M : si riempie in serie ad M7 da Memoria (previa cancellazione).

Il Registro di Funzione presenta permanentemente il proprio contenuto al Decodificatore di Funzione, che opera logicamente sui 16 ingressi, ottenendo 23 uscite (Decodificati Parziali di Funzione) individuanti ognuna una classe di istruzioni
L'unione in And o Or di non più di tre di tali decodificati parziali individua ogni singola istruzione: questa unione viene effettuata negli organi di smistamento logico.

5. DESCRIZIONE DELLE ISTRUZIONI IN FUNZIONE DEI TEMPI.

5.1. ISTRUZIONE MA

Da memoria ad accumulatore senza azzeramento

Codice decimale : 25
codice binario : 0111 1001
decod. parziali :ΔO. ΔA. Δm
configurazione : LdLuImIcIdIuTaFdFu

	0111	1001
Δ O	01..	.001
Δ A	..11	....
Δ m	....	1...

←1 W

Trasferisce (attraverso apposito canale esterno alla tavola di somma di UA) una parola di lunghezza LdLu (con Lu modificata secondo Ta) dalla Memoria, (a partire dalla posizione ImIcIdIu modificata secondo Ta), all'Accumulatore (a partire dalla posizione specificata nel contenuto di Raus P).
Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali.
La parola trasferita è riscritta in Memoria nella sua posizione originale. Se la parola non è segnata, gli indicatori di Acc vengono riposti nello stato $\bar{S}$, N , + .
Se la parola in Memoria è segnata, il segno non entra in Accumulatore e va a disporre i flip flop S e -/+ , il flip flop CN viene comunque riposto in N .
Se si trasferisce un blocco di più parole, ciascuna col proprio segno, il segno della prima è considerato segno generale del blocco.
I segni successivi vengono trasferiti in Acc. senza alterazioni.
Le posizioni di Acc. interessate dal trasferimento vengono cancellate prima di ogni scrittura da Memoria. Anche il piano della Marca è sempre cancellato.
Una Marca viene scritta nell'ultima posizione raggiunta.
La fine dell'operazione avviene nel periodo in cui compare il segnale $\emptyset_C$ di fine di trasferimento che indica la coincidenza fra Cp e RL .

5.1.1. Preparazione MA

5.1.1.1. Ultimo periodo precedente

M 8,0 - RC in R

M 9,0 - disponi alfa
- Sel T

5.1.1.2. Periodo 1

Vedi la fig. 5.1.1.
Si prende l'indirizzo dell'istruzione dal registro ausiliario I, lo si porta in W, gli si somma 10 .
Si estrae il carattere Fu da Memoria e lo si pone Fu (senza transitare attraverso U.A.).
Si cancella RL .

M 0 - Rim. a 01 Cont. p
- Canc. W
- Sel Raus I

M 1 - Disp UAZ

M 1,5 - Int Raus ⟶ W : (è selezionato I da M0)

M 2 - Canc RL

M 3 - Conta +10 in W (1)

M 4 - Ripone D' : serve solo nelle operazioni aritmetiche.

M 4,5 - Pil Mem Int

M 5 - Canc R.F.

M 6 - Sonda Mem
- RC in C

M 7 - M ⟶ Fu

M 7,5 - Pil Mem Scritt
- GMM :

---

1) - Tale conta viene inibita se l'istruzione da preparare segue un'istruzione di salto con condizione verificata.
A tale scopo, R.F. è cancellato solo in α p_1 M5 , per consentire in M3 di inibire +10 .-

M 8,5 - Ripone Φ C (1)
- Sel Raus R (2)

5.1.1.3. Periodo 2

Vedi fig. 5.1.2.
Si estrae il carattere Fd da Memoria e lo si pone in R.F. decine (senza transitare attraverso U.A.). Si copia indietro nel registro di istruzione I , l'indirizzo ottenuto in W dopo una conta -1 .

M 0 - Conta +1 Cp

M 3 - Conta -1 W

M 4,5 - Pil Mem Int

M 6 - Sonda Mem

M 7 - Scritt. W ⟶ Raus : è selezionato I da MO p_1)

M 7,5 - Pil Mem Scritt
- GMM

M 8,5 - M ⟶ Fd

5.1.1.4. Periodo 3

Si estrae da Mem. il carattere Ta, che (sommato a +0) attraversa l'U.A. ed entra nello stat di governo.
Si azzerano gli indicatori di Acc.
αp_3 cancellaz. Raus M

M 0 - Conta +1 Cp
- Selez. Raus O

M 1,5 - Int. Raus ⟶ V : serve solo nella DA

M 3 - Conta -1 W

M 4,5 - Pil Mem Int

---

1) - Può trovarsi già riposto da φ ' al termine della precedente esecuzione.-

2) - E' superfluo per questa istruzione.-

M 5 - Riponi IC - solo per operazioni di cfr.
- Rip. IZ - solo per istr. ΔC o ΔK

M 6 - Sonda Mem

F01 - GMUA : immette in U.A. 1 l'uscita da Mem
- Zero Acc. : immette in U.A. 2 uno zero codice (1)

M 7,5 - Pil Mem Scritt
- GMM

M 8,5 - Ripone S in $\bar{S}$
- Ripone -/+ in +
- Ripone CN in N

5.1.1.5. Periodo 4

Vedi fig. 5.1.3.
Si pone in Vd il carattere Ta estratto da Mem nel p.d.c. precedente. Si completa l'indirizzo di partenza di partenza su Vu contando +1 .
Si estrae la cifra Iu da Mem e la si immette in UA1 . Se il registro Ta esiste si estrae il modificatore corrispondente dalla casella del registro T designato da Ta e la si immette in UA2 . Se non esiste si aziona il generatore di Zeri sul canale da Accumulatore.
Dall'uscita di UA si ottiene Iu (modificata o non) e la si pone in Raus 0 (unità).

M 0 - Conta +1 Cp
- Canc. V (2)

M 1,5 - Canc. Raus M : (0 è stato selezionato ad p_3 MO)

M 3 - Conta -1 W
- UA $\longrightarrow$ Vd : pone in Vd il carattere Ta uscito da UA nel p.d.c. precedente, e immagazzinato nello stat di governo.
- Conta +1 V : Vu passa da zero bit a zero codice
- d $\longrightarrow$ Vu : se nello stat d'uscita della Mem. nel p.d.c. precedente è entrata una delle configurazioni non numeriche: c = 1 , b = 0 , a = 1 oppure: c = 0 , b = 1 , a = 0 , bisogna selezionare uno dei registri T : $, & ,

---

1) - Questo lascia inalterate le configurazioni numeriche: le configurazioni non numeriche (usate per selezionare T$, T& , T£ , T= , T:), vengono alterate come segue: T2, T3, T7, T=, T:, rispettivamente.

2) - Non potendo ottenere l'AND MO p_4 , si è fatto AND fra MO e p_3 ritardato di 1 μs .

£, % , il cui indirizzo iniziale ha un 5 sulle unità. Pertanto si pone un bit 1 in posizione d su Vu (che con la conta +1 passa in definitiva da zero bit a 5).

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α).

M 6 - Sonda Mem
- Sonda Acc

F01 - Zero acc : se su Vd è presente la configurazione non numerica c = 0 , b = 0 , a = 0 ,(1), non si vuole modificare (2) : il decodif. di Acc. non fornisce uscita ed il pilotaggio è inefficace. Si deve azionare il Generatore di Zeri sul canale di Acc.
- GMUA :

M 7 - GUAG :
UA $\longrightarrow$ Raus M : immette l'uscita dallo stat di governo in Raus 0 unità (Raus 0 è stato selezionato in MO di p_4).

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GAA :
- GMM :

5.1.1.6. Periodo 5

Si estrae la cifra Id da Memoria e la si immette in UA 1 . Se il registro Ta esiste, si estrae il modificatore corrispondente dalla 2^ casella del Registro T designato da Ta e la si immette in UA 2 . Se non esiste, si aziona il generatore di zeri sul canale di Accumulatore.
Dall'uscita di UA si ottiene Id (modificata o non) e la si pone in Raus 0 (decine).

M 0 - Conta +1 Cp

---

1) - La stessa condizione inibisce la sonda Acc., ma ciò è superfluo: è stato fatto perchè altre condizioni in OR con questa sono invece indispensabili.

2) - Si può dire che si modifica mediante i registri inesistenti T" oppure T :

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - Zero Acc: vedi corrispondente p_4
- GMUA :

M 7 - GUAG :
- UA $\longrightarrow$ Raus M : immette l'uscita dallo stat di governo in Raus 0 decine (Raus 0 è stato selezionato in M0 di p_4)

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GAA :
- GMM :

5.1.1.7. Periodo 6

Come per il periodo 5, ma riferito alle centinaia di 0
in p6 M5 Rip 0 T solo per istr. Δ T

5.1.1.8. Periodo 7

Come per il periodo 5, ma riferito alle migliaia di 0 . Si sospende il riporto decimale di UA (1) .

5.1.1.9. Periodo 8

Vedi fig. 5.1.4.
Si estrae la cifra Lu da Mem. e la si immette in UA 1 .
Se il registro Ta esiste, si estrae il modificatore corrispondente dalla 5ª casella del registro T designato da Ta, e la si immette in UA 2 .
Se non esiste si aziona il Gen. di zeri sul canale da Acc.

---

1) - Ciò ha lo scopo di evitare che il riporto eventuale generato dalla modifica di Im , vada a sommarsi a Ln nel successivo p.d.c.

Dall'uscita di UA si ottiene Lu (modificato o non) e la si pone nel registro L unità.

M 0 - Conta + 1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - Zero Acc : vedi p_4
- GMUA :

M 7 - GUAG :

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GAA
- GMM

M 9,5 - UA $\longrightarrow$ Lu :

5.1.1.10. Periodo 9

Si estrae la cifra Ld da Mem e la si pone in RL dec., attraverso l'UA, sommando +0 (nonchè l'eventuale riporto della modifica di Lu).
Si ripone α . Si dispone β . Si seleziona l'Accumulatore.

M 0 - Conta +1 Cp
- Canc. V (1): è superfluo per questa istruzione

M 3 - Conta -1 W
- UA $\longrightarrow$ Vd : è superfluo per questa istruzione
- d $\longrightarrow$ Vu : è superfluo per questa istruzione

M 4,5 - Pil Mem Int

M 6 - Sonda Mem
- RC in R

---

1) - Non potendo ottenere l'AND MO . p9 , si è fatto AND fra MO e p8 ritardato di 1 μs .

F01 - Zero Acc : immette in UA 2 uno zero codice
- GMUA :

M 7 - GUAG :

M 7,5 - Pil Mem Scritt
- GMM

M 8 - Disp. IC : solo per confronti e se si ha SA .

M 8,5 - Dispone β
- Ripone α

M 9,5 - UA $\longrightarrow$ Ld

5.1.2. Esecuzione MA

5.1.2.1. Periodo 1

Vedi fig. 5.1.5. e 5.1.6.
Si porta l'indirizzo dell'operando dal Raus O a W (1).
Si porta l'indirizzo di partenza dell'Acc. da Raus P a V e lo si copia poi di nuovo indietro in Raus P .
Se si estrae il primo carattere da Mem. e lo si pone nella prima posizione dell'Acc. attraverso il canale esterno alla tavola di somma, se è un segno, si dispongono di conseguenza flip flop S e -/+ .
Si confronta Cp con RL .

M O - Rim. a O1 Cp
- Canc. W
- Canc. V
- Sel Raus P

M 1 - Disp. D se +/- è in - .
- Disp. UAZ

M 1,5 - Int Raus ⟶ W : è selezionato O da MO p3 .
- Int Raus ⟶ V : è selezionato P da MO

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato A dall'inizio di β .

M 5 - Scritt V ⟶ Raus A : è selezionato P da MO

M 5,5 - Compara (2)

M 6 - Sonda Mem
- Sonda Acc (3)
- RC in C

FO1 - GMUA (3)

M 7 - GMA
- Scritt W ⟶ Raus (3)

---

1) - e lo si copia di nuovo indietro in Raus O, ma ciò serve solo in Moltiplicazione.
2) - La comparazione avviene anche in ogni p.d.c. di α , ma solo in β si può raggiungere la coincidenza.
3) - E' superfluo.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt
- GAA : solo se da Mem è uscito un segno (1)

M 8 - Dispone S : solo se da Mem è uscito un segno
- Dispone -/+ : solo se da Mem è uscito un segno .

5.1.2.2. Periodo 2

Si estrae il secondo carattere da Mem.
Se il primo fu un segno, si scriva questo secondo nella prima posizione di Acc, (cancellando quanto scrittovi nel p.d.c. 1).
Se il primo non fu un segno, si scrive il secondo carattere uscente da Mem nella seconda posizione di Acc, sempre attraverso il canale esterno alla tavola di somma.
Si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V : solo se nello stat d'uscita Mem non c'è un segno (uscito ad M6 di p1).

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc (2)

F01 - GMUA (2)

M 7 - GMA

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

---

1) - E' previsto per le operazioni aritmetiche (ΔK): per la presente istruzione conduce a scrivere nella prima casella di Acc una configurazione priva di senso, perchè è aperto anche il canale GMA . Ciò non disturba perchè in p2 , se vi fu segno da Mem in p1, si chiama la stessa cella di Acc.-

2) - E' superfluo.-

5.1.2.3. Periodo 3

Si estrae il successivo carattere da Mem e lo si scrive in Acc di seguito al carattere scritto nel p.d.c. precedente, sempre attraverso il canale esterno alla tavola di somma.
Si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda acc (1)

FO1 - GMUA (1)

M 7 - GMA

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

I successivi periodi sono tutti uguali a p3, fino al periodo in cui il contenuto di Cp coincide con contenuto di RL : la comparazione dà allora esito favorevole ed il periodo in cui ciò avviene è l'ultimo.

5.1.2.4. Ultimo periodo

Si estrae l'ultimo carattere da Mem e lo si scrive in Acc di seguito al carattere scritto nel precedente p.d.c., sempre attraverso il canale esterno alla tavola di somma.
Si confronta Cp con RL .
Si scrive la Marca in Acc.
Si ripone α . Si dispone β .

M 0 - Conta +1 Cp

---

1) - E' superfluo.-

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disponi $\emptyset_c$: nasce φ
- Poni RC in R

M 6 - Sonda Mem
- Sonda Acc (1)

FO1 - GMUA (1)

M 7 - GMA
- Scrivi Marca

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM

M 8,5 - Riponi β
- Sel T
- Riponi $\emptyset_c$

M 9 - Disponi α .

---

1) - E' superfluo.

5.2. ISTRUZIONE MoA

Da memoria ad accumulatore con azzeramento

Codice decimale : 20
Codice binario : 0111 0001
Decod. Parziali : Δ 0. ΔA. Δp
Configurazione : LdLuImIcIdIuTaFdFu

	0111	0001
Δ 0	01..	.001
Δ A	..11	....
Δ p	....	0...

← - 1 W

Come per la MA, con la differenza che la parola trasferita non viene riscritta in Mem. Le posizioni di Mem interessate dal trasferimento restano azzerate, poichè in tali posizioni si scrive un bit 1 sul piano a .

5.2.1. Preparazione MoA

Come per la preparazione MA .

5.2.2. Esecuzione MoA

Come per l'esecuzione MA, con la differenza che GMM non si apre mai, e ad ogni p.d.c. sul piano a di Mem si scrive un bit 1 .

5.3. ISTRUZIONE AM

Da accumulatore a Memoria senza azzeramento

Codice decimale : 26
Codice binario : 0111 1011
Decod. Parziali : $\Delta 1 . \Delta A . \Delta m$
Configurazione : LdLuImIcIdIuTaFdFu

	0111	1011
Δ 1	01..	.011
Δ A	..11	....
Δ m	....	1...

← - 1 W

Trasferisce una parola di lunghezza LdLu dall'Accumulatore (a partire dalla posizione specificata dal contenuto di Raus P), alla Memoria (a partire dalla posizione ImIcIdIu modificata secondo Ta). Sia l'istruzione, che il contenuto del registro Ta, vengono riscritti nelle loro posizioni originali. A seconda dello stato degli indicatori di Acc possono presentarsi vari casi:

1) - Se il contenuto di Acc non è segnato, il contenuto della prima posizione di Acc si scrive senza alterazioni nella prima posizione di Mem, e così via, indipendentemente dallo stato del flip flop CN .

2) - Se il contenuto di Acc è segnato, nella prima posizione di Mem si scrive il segno; nella seconda posizione di Mem si scrive il contenuto della prima casella di Acc, nella terza posizione di Mem si scrive il contenuto della seconda posizione di Acc e così via.

2A) - Se il contenuto di Acc è segnato, ma normale, segni e cifre vengono trasferiti senza alterazioni.

2B) - Se invece il contenuto di Acc è segnato e in complemento, il segno viene scritto in Mem cambiato, ed inoltre si complementano a 10 le cifre uscenti da Acc fino alla prima cifra significativa inclusa. Le cifre successive vengono complementate a 9 .

La parola trasferita è riscritta in Acc nella sua posizione originale.
Lo stato degli indicatori di Acc non viene alterato. La posizione delle marche non viene alterata.
Le posizioni di Mem interessate dal trasferimento vengono cancellate prima di ogni scrittura da Acc, sopprimendo la sonda di Memoria.
La fine dell'operazione avviene nel periodo in cui compare il segnale $\emptyset_C$ di fine di trasferimento, che indica la coincidenza fra Cp e RL .

5.3.1. Preparazione AM

Come per la preparazione MA, con la differenza che nel p.d.c. 3 non vengono azzerati gli indicatori di Acc.

5.3.2. Esecuzione AM

5.3.2.1. Periodo 1

Si porta l'indirizzo dell'operando dal Raus O a W . (2g).
Si porta l'indirizzo di partenza dell'Acc da Raus P a V e lo si copia poi di nuovo indietro in Raus P .

Se non si ha segno in Acc, si estrae il primo carattere di Acc e lo si pone nella prima posizione di Mem.

Se si ha segno in Acc, non si pilota Acc: un generatore di segno scrive nella prima posizione di Mem il segno contenuto nel flip-flop -/+ , se l'Acc. è in normale, scrive invece il segno opposto se l'Acc è in complemento.
Si confronta Cp con RL .

M 0 - Rim. a 01 Cp
- Canc. W
- Canc. V
- Sel Raus P

M 1,5 - Int Raus ⟶ W : è selezionato O da α p3 MO
- Int Raus ⟶ V : è selezionato P da MO

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato A dall'inizio di β): solo se non c'è segno in Acc .

M 5 - Scritt V ⟶ Raus A : è selezionato P da MO

M 5,5 - Compara (1 g)

M 6 - Sonda Acc (1)
- RC in C

F01 - GMUA (2)

M 7 - GAM aperto in β Δ 1
- Gen segno: inietta su GAM il segno vero se l'Acc è N, invertito se l'Acc è C
- Scritt W ⟶ Raus (2)

M 7,5 - Pil Mem Scritt
- GMM (3)
- Pil Acc. Scritt : **solo se non v'è segno in Accumulatore**

---

1) - Se l'Acc. è segnato, è superfluo. 2) - E' superfluo.
3) - E' superflua, in quanto manca la sonda.

GAA (1)

M 9 - Riponi U.A.Z.: solo da Acc è uscita una cifra significat.
Se Acc è segnato, non è uscito nulla.

5.3.2.2. Periodo 2

Se non si ha segno in Acc, si estrae il secondo carattere di Acc e lo si pone nella seconda posizione di Mem.
Se si ha segno in Acc, si estrae il primo carattere di Acc e lo si pone nella seconda posizione di Mem senza alterarlo, se l'Acc è N, complementandolo invece a 10 se l'Acc è C .
Si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V : solo se non c'è segno in Acc

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Acc

F 01 - GMUA (2)

M 7 - GAM
- Compl. Acc (3) : solo se Acc è segnato e C
- +1 A : solo se Acc è segnato e C

M 7,5 - Pil Mem Scritt
- GMM (2)
- Pil Acc Scritt
- GAA
- Rig marca

5.3.2.3. Periodo 3

Si estrae il successivo carattere da Acc e lo si pone in Mem di seguito al carattere scritto nel pdc precedente: se Acc è non se-

---

1) - Se l'Acc. è segnato, è superfluo.
2) - E' superflua .
3) - Si ricordi che il complementatore è a 9 .

gnato o normale, non lo si altera; se Acc è segnato e complemento, lo si complementa. La complementazione è a 9 se in uno dei pdc precedenti è uscita una cifra significativa; in caso contrario è a 10 .
Si confronta Cp con RL

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Acc

F01 - GMUA (1)

M 7 - GAM
- Compl Acc: solo se Acc è segnato e C
- +1 A : solo se Acc è segnato e C, e UAZ=UAZ

M 7,5 - Pil Mem Scritt
- GMM (2)
- Pil Acc Scritt
- GAA
- Rig Marca

I successivi periodi sono tutti uguali a p3 fino al periodo in cui il contenuto di Cp coincide col contenuto di RL : la comparazione dà allora esito favorevole ed il periodo in cui ciò avviene è l'ultimo.

5.3.2.4. Ultimo periodo

Si estrae l'ultimo carattere da Acc e lo si pone in Memoria di seguito al carattere scritto nel pdc precedente, senza alterarlo se Acc è senza segno o normale, complementando (a 10 o a 9 con le reg. date sopra) se Acc è segnato e complemento.
Si confronta Cp con RL .
Si ripone β . Si dispone α .

---

1) - Superflua.
2) - Superflua perchè manca la sonda.

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disponi $\emptyset_c$: nasce φ
- Poni RC in R

M 6 - Sonda Acc

FO1 - GMUA (1)

M 7 - GAM
- Compl. Acc: solo se Acc è segnato e C
- +1 A : solo se Acc è segnato e C, e UAZ = UAZ

M 7,5 - Pil Mem Scritt
- GMM (2)
- Pil Acc Scritt
- GAA
- Rig Marca

M 8,5 - Riponi β
- Sel T
- Rip $\emptyset_c$

M 9 - Disponi α .

---

1) - E' superflua.
2) - E' superflua, perchè manca la sonda.

5.4. ISTRUZIONE AoM

Da Accumulatore a Memoria con Azzeramenti

Codice decimale : 21
Codice binario : 0111 0011
Decod. Parziali : Δ 1. ΔA. Δ p
Configurazione : LdLuImIcIdIuTaFdFu

	0111	0011
Δ 1	01..	.011
Δ A	..11	....
Δ p	....	0...

← -1 W

Come per la AM, con la differenza che la parola trasferita non viene riscritta in Acc. Le posizioni di Acc interessate dal trasferimento restano pertanto azzerate e munite di marca. Gli indicatori di Acc vengono riposti nello stato $\bar{S}$, N , + .

5.4.1. Preparazione AoM

Come per la preparazione MA .

5.4.2. Esecuzione AoM

Come per l'esecuzione AM, con le seguenti differenze:

GAA non si apre mai .
Ad ogni pdc si scrive un bit 1 sui piani a ed f del M_{od}.
Nell'ultimo pdc, a M8,5 si ripongono S , CN , -/+ .

5.5. ISTRUZIONE MT

Da memoria a modificatore senza azzeramento

Codice decimale : 35
Codice binario : 0110 1001
Decod. Parziali : Δ0 ΔT Δm
Configurazione : LuTbImIcIdIuTaFdFu

	0110	1001
Δ 0	01..	.001
Δ T	..10	....
Δ m	....	1...

← - 1 W

Trasferisce (attraverso apposito canale esterno alla tavola di somma di UA) una parola di lunghezza Lu dalla memoria (a partire dalla posizione ImIcIdIu modificata secondo Ta) al modificatore (a partire dalla posizione Tb modificata secondo Ta).

Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali. La parola trasferita è riscritta in memoria nella sua posizione originale. Se la parola in mem. è segnata, il segno viene trasferito.

Le posizioni di Tb interessate dal trasferimento vengono cancellate prima di ogni scrittura da mem.

Una Marca viene scritta nell'ultima posizione raggiunta.

La fine dell'operazione avviene nel periodo in cui compare il segnale $\emptyset_C$ di fine di trasferimento, che indica la coincidenza tra Cp e RL .

5.5.1. Preparazione MT

Periodi da 1 a 7 (incluso), come per la MA, con la differenza che nel pdc 3 non vengano azzerati gli indicatori di acc. Inoltre a p6 M8 si ripone OT (serve solo in operaz. aritm. su registro T).

5.5.1.1. Periodo 8

Vedi fig. 5.5.1.
Si estrae il carattere Tb da Mem e lo si immette in UA 1 .

Se il registro Ta esiste, si estrae il modificatore corrispondente dalla quinta casella del registro T individuato da Ta, e lo si immette in UA 2 . Se non esiste, si aziona il generatore di zeri sul canale da Acc.

Dall'uscita di UA si ottiene Tb (modificato o non) e lo si pone nello stat di governo.

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di)

M 6 - Sonda Mem
- Sonda Acc

FO1 - Zero Acc: vedi p4 (pag. 5)
- GMUA

M 7 - GUAG

M 7,5 - Pil Mem scritt
- Pil Acc scritt
- GAA
- GMM

5.5.1.2. Periodo 9

Vedi fig. 5.5.2.
Si pone in Vd il carattere Tb estratto da Mem. nel pdc precedente e immagazzinato nello stat di governo. Si estrae la cifra Lu da Mem e la si pone in RL dec. attraverso UA, sommando +0 (nonchè l'eventuale riporto della modifica di Lu).
Si ripone α. Si dispone β. Si ripone l'indicatore di trabocco in RT, OT .

M 0 - Conta +1 Cp
- Canc V (1)

M 3 - Conta -1 W
- UA $\longrightarrow$ Vd : pone in Vd il carattere Tb uscito da UA nel pdc precedente e immagazzinato nello stat di governo.

---

1) - Non potendo ottenere l'AND MO p9, si è fatto AND fra MO e p8 ritardando di 1 μs . Non si può cancellare prima, perchè V in p8 sta selezionando l'ultima posizione di Ta . Non si può cancellare dopo, perchè a M3 si trasferisce in Vd da UA .

- d ⟶ Vu : se nello stat d'uscita della Mem. nel pdc precedente entrata una delle configurazioni non numeriche: c = 1 , b = 0 , a = 1 , oppure c = 0 , b = 1 , a = 0 , bisogna selezionare uno dei registri T : $, & , £ , % , il cui indirizzo iniziale ha un 5 sulle unità. Pertanto si pone un bit 1 in posizione d su Vu . La conta +1 che avverrà in β p1 porterà a 1 il bit a di Vu .

M 4,5 - Pil Mem Int

M 6 - Sonda Mem
- RC in R

F01 - Zero Acc : immette in UA2 uno zero codice
- GMUA

M 7 - GUAG

M 7,5 - Pil Mem Scritt
- GMM

M 8,5 - Disponi β

M 9 - Riponi α

M 9,5 - UA ⟶ Lu

5.5.2. Esecuzione MT

5.5.2.1. Periodo 1

Si porta l'indirizzo dell'operando da Raus 0 a W (2g) .
Si completa l'indirizzo di partenza su Vu, contando +1 .

Si estrae il primo carattere da Mem e lo si scrive nella prima posizione di Tb attraverso il canale esterno alla tavola di somma.
Si confronta Cp con RL .

M 0 - Rim a 01 Cp
- Canc W
- Sel Raus P : (1)

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 MO

M 2 - zero RL dec (2)

---

1) - E' superfluo.
2) - Consente la comparazione con Cp .

M 3 - Conta +1 V

M 4,5 - Compara (1g)

M 6 - Sonda Mem
- Sonda Acc (1)
- RC in C

FO1 - GMUA (1)

M 7 - GMA
- Scritt W ⟶ Raus (1)

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

5.5.2.2. Periodo 2

Vedi fig. 5.5.3.
Si estrae la seconda cifra da Mem e la si scrive nella seconda posizione di Tb . Si fa uso del canale esterno alla tavola di somma. Si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di α.

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc (1)

FO1 - GMUA (1)

M 7 - GMA

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

I periodi dal terzo in poi sono uguali ai corrispondenti dell'esecuzione MA, ove sia selezionato T invece di A .

---

1) - E' superfluo.

5.6. ISTRUZIONE MoT

Da Memoria a RT con Azzeramento

Codice Decimale : 30
Codice binario : 0110 0001
Decod. Parziali : Δ0 ΔT Δp
Configurazione : LuTbImIcIdIuTaFdFu

	0110	0001
Δ0	01..	.001
ΔT	..10	....
Δp	....	0...

← - 1 W

Come per la MT, con la differenza che la parola trasferita non viene riscritta in Mem. Le posizioni di Mem interessate dal trasferimento restano azzerate, poichè in tali posizioni si scrive un bit 1 sul piano a .

5.6.1. Preparazione MoT

Come per la preparazione MT .

5.6.2. Esecuzione MoT

Come per l'esecuzione MT, con la differenza che GMM non si apre mai, e ad ogni pdc sul piano a di Mem si scrive un bit 1 .

5.7. ISTRUZIONE TM

Da RT a Memoria senza azzeramento

Codice Decimale : 36
Codice Binario : 0110 1011
Decod. Parziali : Δ 1 Δ T Δ m
Configurazione : LuTbImIcIdIuTaFdFu

	0110	1011
Δ 1	01..	.011
Δ T	..10	....
Δ m	....	1...

←——
-1 W

Trasferisce una parola di lunghezza Lu da RT (a partire dalla posizione Tb modificata secondo Ta), alla Memoria (a partire dalla posizione ImIcIdIu modificata secondo Ta). Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali.
La parola trasferita è riscritta nel RT nella sua posizione originale, senza alterare la posizione delle marche.
Le posizioni di Mem interessate dal trasferimento vengono cancellate prima di ogni scrittura da Acc., sopprimendo la sonda di Memoria. La fine dell'operazione avviene nel pdc in cui compare il segnale $\emptyset_c$ di fine di trasferimento, che indica la coincidenza fra Cp e RL .

5.7.1. Preparazione TM

Come per la preparazione MT .

5.7.2. Esecuzione TM

5.7.2.1. Periodo 1

Vedi fig. 5.7.1.
Si porta l'indirizzo dell'operando da Raus 0 a W (1)
Si completa l'indirizzo di partenza su V , contando +1 .

---

1) - E lo si copia di nuovo indietro in Raus 0, ma ciò serve solo in moltiplicazione.

Si estrae il primo carattere dal Modificatore Tb e lo scrive nella prima posizione di Mem.
Si confronta Cp con RL .

M 0 - Rim a 01 Cp
- Canc. W
- Sel Raus P (1)

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 MO

M 2 - Zero RL_{dec} (2)

M 3 - Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di α

M 5,5 - Compara (1g)

M 6 - Sonda Acc
- RC in C

F01 - GMUA (1)

M 7 - GAM
- Scritt W ⟶ Raus (1)

M 7,5 - Pil Mem Scritt
- GMM (3)
- Pil Acc Scritt
- GAA
- Rig Marca

5.7.2.2. Periodo 2

Si estrae il successivo carattere dal Mod. Tb e lo si scrive nella successiva posizione di Mem .
Si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

---

1) - E' superfluo.
2) - Consente la comparazione con Cp
3) - E' superfluo, in quanto manca la sonda.

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T

M 5,5 - Compara

M 6 - Sonda Acc

FO1 - GMUA (1)

M 7 - GAM

M 7,5 - Pil Mem Scritt
- GMM (2)
- Pil Acc Scritt
- GAA
- Rig Marca

I successivi periodi sono tutti uguali a p2, fino al pdc in cui il contenuto di RL coincide col contenuto di Cp : la comparazione dà allora esito favorevole e il pdc in cui ciò avviene è l'ultimo.

5.7.2.3. Ultimo periodo

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T

M 5,5 - Compara
- Disponi $\emptyset_c$: nasce φ
- Poni RC in R

M 6 - Sonda Acc

FO1 - GMUA (1)

M 7 - GAM

M 7,5 - Pil Mem Scritt
- GMM (2)
- Pil acc scritt
- GAA
- Rig Marca

M 8,5 - Riponi β
- Sel T
- Riponi $\emptyset_c$

M 9 - Disponi

---

1) - E' superfluo. 2) - E' superfluo, in quanto manca la sonda.

5.8. ISTRUZIONE ToM

Da RT a Memoria con Azzeramento

Codice Decimale: 31
Codice Binario : 0110 0011
Decod.Parziali : Δ1 ΔT Δp
Configurazione : LuTbImIcIdIuTaFdFu

	0110	0011
Δ 1	01..	.011
Δ T	..10	....
Δ p	....	0...

←
-1 W

Come per la TM, con la differenza che la parola trasferita non viene riscritta nel RT . Le posizioni di RT interessate dal trasferimento restano pertanto azzerate e munite di Marca.

5.8.1. Preparazione ToM

Come per la preparazione MT .

5.8.2. Esecuzione ToM

Come per l'esecuzione TM, con le seguenti differenze:
GAA non apre mai.
Ad ogni pdc si scrive un bit 1 sui piani a ed f del RT .

5.9. ISTRUZIONE IT

Da istruzione a T

Codice decimale : 45
codice binario : 0100 1001
Ddcod.parziali : Δ0. ΔI. Δm
Configurazione : X L $I_m I_c I_d I_u T_a F_d F_u$

	0100	1001
Δ 0	01..	.001
Δ I	..00	....
Δ m	....	1...

$\xleftarrow{-1\ W}$

Trasferisce i caratteri L I_m I_c I_d I_u dell'istruzione dalla Memoria al registro Ta del RT, attraverso il canale esterno alle tavole di UA.
Non è modificabile.
E' priva di fase esecutiva.
I caratteri dell'istruzione vengono riscritti in Mem. nelle loro posizioni originali. Le posizioni del RT interessate dal trasferimento vengono cancellate prima di ogni scrittura da Memoria, chiudendo il canale di rigenerazione.
Il carattere X serve solo a completare i 9 p.d.c. di preparazione.

5.9.1. Preparazione ed esecuzione IT

Periodi 1 e 2 come per la preparazione MA

5.9.1.1. Periodo 3

Si estrae da Mem il carattere Ta, che (sommato + 0) attraversa l'U.A. ed entra nello stat di governo.

M 0 - Conta +1 Cp

M 3 - Conta -1 W

M4,5 - Pil Mem Int

M 6 - Sonda Mem

F01 - GMUA
- Zero Acc : vedi nota (1) pag.

M7,5 - Pil Mem Scritt
- GMM

5.9.1.2. Periodo 4 - Vedi fig. 5.9.1.

Si pone in V_d il carattere T_a estratto da Mem. nel pdc precedente. Si estrae la cifra I_u da Mem e la si pone nella prima posizione del registro Ta del RT (1).

M 0 - Conta +1 Cp
- Canc. V : vedi nota 2 a pag. 71

M 1,5 - Canc. Raus M : (è selezionato 0 da α p_3 MO) (2)

M 3 - Conta -1 W
- UA $\longrightarrow$ v_d : pone in v_d il carattere uscito da UA nel pdc precedente e immagazzinato nello stat di governo. Vedi nota a pag.
- Conta +1 V : Vu passa da zero bit a zero codice
- d $\longrightarrow$ Vu : vedi pag. 71 p4

M4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA (2)

M 7 - GUAG (2)
- UA $\longrightarrow$ Raus M (2)

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM
- GMA

5.9.1.3. Periodo 5 - Vedi fig. 5.9.2.

Si estrae la cifra Id da Memoria e la si pone nella seconda posizione del registro Ta del RT (3).

M 0 - Conta +1 Cp

M 3 - Conta - 1 W
- Conta + 1 V

---

1) - La cifra I_u (alterata dal passaggio attraverso UA che somma l'eventuale contenuto di Ta) viene immessa anche nel Raus 0 unità, ma ciò è superfluo.
2) - E' superfluo.
3) - La cifra Id (alterata dal passaggio attraverso UA, che somma l'eventuale contenuto di Ta) viene immessa anche in Raus 0 decine.

M 4,5 - Pil Mem Int
- Pil Acc Int: (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

FO1 - GMUA (1)

M 7 - GUAG (1)
- UA $\longrightarrow$ Raus M (1)

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM
- GMA

Periodo 6

Come per il periodo 5, ma riferito alle centinaia di I .

Periodo 7

Come per il periodo 5, ma riferito alle migliaia di I (2). Inoltre nasce un segnale φ , che serve solo nella CIT .

Periodo 8

Come per il periodo 5, ma riferito a L (3)

Periodo 9 (superfluo)

Si estrae il carattere X (superfluo) da Mem e lo si pone in RL dec. attraverso UA, sommando +0 (nonchè l'eventuale riporto della modifica spuria, secondo nota (3), di Lu).
Si ripone RC .

---

1) - E' superfluo.
2) - Si sopprime il riporto decimale di UA, ma ciò è superfluo.
3) - La cifra L (alterata dal passaggio attraverso UA che somma l'eventuale contenuto di Ta) viene immessa anche in RL unità, ma ciò è superfluo.-

5.10. ISTRUZIONE +MA

Addiziona Memoria ad Accumulatore, ivi

Codice Decimale: 22
Codice binario : 0111 0111
Decod. parziali: $\Delta 2$. ΔA. Δp
Configurazione : $L_d L_u I_m I_c I_d I_u T_a F_d F_u$

	0111	0111
$\Delta 2$	01..	.111
ΔA	..11	....
Δp	....	0...

$\longleftarrow$
-1 W

Somma un numero di lunghezza LdLu (Lu modificato secondo Ta), proveniente da Memoria, a partire dalla posizione ImIcIdIu (modificata secondo Ta), con un numero in Acc., a partire dalla posizione specificata dal contenuto del Raus P, di lunghezza fissata dalla marca. Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali. Il risultato viene scritto in Acc. Il numero in Mem viene riscritto nella sua posizione originale. Il numero da Mem può essere segnato o non segnato.

L'Acc può essere segnato o non segnato: se è segnato, può essere normale o in complemento.

Se uno dei numeri è segnato, il risultato è segnato.

Il segno di Acc rimane comunque invariato: eventualmente si aziona il complementatore sul canale da Memoria, e/o si agisce (al termine dell'operazione), sul flip-flop CN, secondo le regole esposte nella parte II^.

La fine dell'operazione avviene nel pdc in cui compare il segnale φ' di fine di somma (o differenza) generato secondo i criteri esposti nella parte II^. Se uno dei numeri da sommare termina prima del segnale φ', si aziona il corrispondente generatore di zeri.
Per quanto riguarda la marca, se essa non viene letta nel corso dell'operazione, non viene spostata. Se è stata letta, viene scritta nel pdc in cui compare il segnale φ'.

Ad ogni pdc dell'esecuzione si esamina l'uscita da UA : se è diversa da zero si ripone il flip-flop IZ . Se al termine dell'esecuzione IZ è ancora disposto, il risultato è nullo (almeno limitatamente alla lunghezza LuLd).

5.10.1. Preparazione +MA

Come per la preparazione MA, con le differenze seguenti: al pdc 3 gli indicatori di Acc non vengono azzerati, l'indicatore di zero IZ viene disposto al pdc 3 , M 5 .
Si osservi che ora la disposizione del flip-flop D sarà efficace dato che in fase esecutiva il segnale da Mem transita attraverso il complementatore.

5.10.2. Esecuzione +MA

5.10.2.1. Periodo 1 - Vedi fig. 5.10.1.

Si porta l'indirizzo dell'operando da Raus O a W (1).
Si porta l'indirizzo di partenza dell'Acc da Raus P a V e lo si copia poi di nuovo indietro in Raus P .
Si estrae il primo carattere da Mem e lo si immette in UA_1 (complementato o no a 10).
Si estrae la prima cifra da Acc e la si immette in UA_2 .
Se il carattere da Mem è un segno, si dispone il flip-flop S, si blocca l'uscita da UA verso Acc, e si rigenera Acc.
Se non è un segno, si pone l'uscita da UA nella prima posizione di Acc. L'eventuale riporto dispone +1 Mem per il prossimo pdc.
Si confronta Cp con RL .

M O - Rim e O1 Cp
- Canc W
- Canc V
- Sel Raus P
- Rip UMZ (2)

M 1 - Disponi UMZ
- Disp. UAZ

M 1,5 - Int Raus ⟶ W : è selzionato O da α p3 MO
- Int Raus ⟶ V : è selezionato P da MO

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato A dall'inizio di β .

M 5 - Scritt. V ⟶ Raus A : è selezionato P da MO

---

1) - e lo si copia di nuovo indietro in Raus O, ma ciò serve solo in moltiplicazione.
2) - E' superfluo in questo pdc .

M 5,5 - Compara (1g)

M 6 - Sonda Mem
- Sonda Acc
- RC in C

F01 - GMUA
- GAUA

M 7 - Riponi IZ : se da UA è uscita una cifra significativa.
- Scritt W ⟶ Raus (1)
- Compl Mem : solo se D è disposto.
- +1 : se D è disposto.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt
- GAA : se da Mem è uscito un segno

M 8,5 - Disponi S : se da Mem è uscito un segno
- Disponi SM4 : se da Mem è uscito un segno
- Rip +1 Mem.

5.10.2.2. Periodo 2 - Vedi fig. 5.10.2.

Si estrae la seconda cifra da Mem e la si immette in UA_1, complementando o no, (a 10 se nel precedente pdc era uscito zero o segno da Mem, a 9 se era uscita una cifra significativa).

Se il primo carattere estratto da Mem fu un segno, si estrae di nuovo la prima cifra da Acc e la si immette in UA_2 . Indi si pone l'uscita di UA nella prima posizione di Acc. Se il primo carattere estratto da Mem non fu un segno, si estrae la seconda cifra da Acc e la si immette in UA_2 . Indi si pone l'uscita di UA nella seconda posizione di Acc.
Si confronta Cp con RL .

M 0 - Conta +1 Cp
- Rip UMZ : solo se nel pdc precedente è uscita una cifra significativa.

M 3 - Conta -1 W
- Conta +1 V : solo se nello stat d'uscita di Mem non c'è segno, (uscito a M6 di p1).

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

---

1) - Serve solo in moltiplicazione.

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

M 7 - GUAA
- Compl Mem : se D è disposto
- +1 Mem : se D e UMZ oppure +1 sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

5.10.2.3. Periodo 3

Si estrae la successiva cifra da Mem e la si immette in UA_1, complementando o no (a 9 se in uno dei pdc precedenti è uscita una cifra significativa, a 10 negli altri casi).
Si estrae la successiva cifra da Acc e la si immette in UA_2 .
Si pone l'uscita di UA nella successiva posizione di Acc.

Si confronta Cp con RL .

M 0 - Conta +1 Cp
- Rip UMZ: se nel precedente pdc è uscita una cifra significativa.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

M 7 - GUAA
- Compl Mem : se D è disposto
- +1 Mem : se D e UMZ sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

I successivi periodi sono tutti uguali a p3 , fino al periodo in cui ha inizio la procedura di fine.

5.10.3. Procedura di fine

5.10.3.1. Procedura di fine tipo 1

Il numero da Mem è più lungo del numero in Acc .

5.10.3.1.1. Periodo m

Come in periodo 3, e inoltre: esce la marca da Acc. Si dispone Ø A .

M 6 - Esce f_A

M 8,5 - Disp Ø A

5.10.3.1.2. Periodo m+1

Si estrae la successiva cifra da Mem e la si immette in UA_1 , complementando etc.
Si pilota Acc senza sonda.
Si aziona il gen. di zeri sul canale da Acc. complementando a 9 gli zeri se Acc è C .
Si pone l'uscita di UA nella successiva posizione di Acc
Si confronta Cp con RL .

M 0 - Conta +1 Cp
- Rip UMZ : se nel precedente pdc è uscita una cifra significativa.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Mem

F01 - Zero Acc : immette in UA_2 zero codice
- GMUA

M 7 - GUAA
- Compl Acc : solo se CN è in C
- Compl Mem : se D' è disposto
- +1 Mem : se D' e UMZ sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

I successivi periodi sono tutti uguali al m+1 esimo fino a che si presenta uno dei due sottocasi seguenti.

5.10.3.1.3. Sottocaso 1 A

Periodo finale

Come per il pdc n+1 ed inoltre: la comparazione dà esito favorevole: nasce $\emptyset_C$, indi $\emptyset_M$. R e D sono in condizioni tali da consentire la nascita immediata di φ' .
Si ripongono β , $\emptyset_C$, $\emptyset_A$, $\emptyset_M$
Si dispone α , si sel T
Si scrive la marca in Acc

M 0 - Conta +1 Cp
- Rip UMZ : se nel precedente pdc è uscita una cifra significativa.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disponi $\emptyset_C$

M 6 - Disp $\emptyset_M$
- Sonda Mem

F01 - Zero Acc
- GMUA

M 7 - GUAA
- Compl Acc: solo se CN è in C
- Compl Mem: se D' è in D'
- +1 Mem : se D' e UMZ sono disposti
- Rip IZ : se da UA è uscita una cifra significativa.

M 7,5 - Disp φ'

M 8 - poni RC in R

M 8,5 - Riponi $\emptyset_A$ $\emptyset_M$
- Pil Mem Scritt
- GMM
- Pil Acc Scritt
- scrivi Marca
- Riponi β
- Riponi $\emptyset_c$

M 9 - Disp α
- Sel T

M 9,5 - Riponi φ'

5.10.3.1.4. Sottocaso 1 B

Periodo prefinale

Come per il pdc n+1 esimo ed inoltre:
La comparazione dà esito favorevole: nasce $\emptyset_c$ indi $\emptyset_M$. R e D non consentono la nascita di φ'.

M 0 - Conta +1 Cp
- Rip UMZ : se nel periodo precedente è uscita una cifra significativa.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disponi $\emptyset_c$

M 6 - Disp $\emptyset_M$

F01 - Zero acc
- GMUA

M 7 - GUAA
- Compl. Acc : solo se CN è in N
- Com.Mem: se D' è in D'
- +1 Mem : se D' e UMZ sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

Periodo finale

Nasce $\emptyset_{M1}$
Non si pilota Mem. Si aziona invece il gen. di zeri sul canale da Mem. Il complementatore sul canale da Mem trasforma eventualmente gli zeri in nove. Si pilota Acc senza sonda. Si aziona il gen. di zeri sul canale da Acc, complementando gli zeri a 9 se CN è in C . Si pone l'uscita da UA nelle successive posizioni di Acc.
Nasce φ' . Si ripongono β , $\emptyset_C$, $\emptyset_A$, $\emptyset_M$, $\emptyset_{M1}$.
Si disp α si sel T . Si scrive la marca in Acc.

M 0 - Conta +1 Cp
- Riponi UMZ : se nel pdc precedente è uscita una cifra significativa.

M 3 - Conta -1 W (1)
- Conta +1 V

M 3,5 - Disp $\emptyset_{M1}$

M 4,5 - Pil Acc Int

M 5,5 - Compara (2)

M 6 - Sonda Mem (2)

F01 - Zero Acc
- Zero Mem
- GMUA

M 7 - GUAA
- Compl Acc : solo se CN è in C .
- Compl Mem : se D è in D
- +1 Mem : se D e UMZ sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa.

M 7,5 - Pil Acc Scritt
- GMM (2)
- Scrivi Marca
- Disp φ'

M 8 - Poni RC in R

---

1) - E' superfluo, perchè manca il pilotaggio.
2) - E' superfluo.

M 8,5 - Riponi $\emptyset_A$, $\emptyset_M$, $\emptyset_{M1}$
- Riponi β , $\emptyset_c$
- Disp UAZ

M 9 - Disp α
- Sel T

M 9,5 - Rip φ'

5.10.3.2. Procedura di fine tipo 2

Il n° in acc. è di lunghezza eguale al numero in memoria.

5.10.3.2.1. Sottocaso 2 A

Periodo finale

Come per il pdc 3 ed inoltre:
La comparazione dà esito favorevole, nasce $\emptyset_c$, indi $\emptyset_M$ e, ad M6 esce f_a . Le condizioni R e D consentono la nascita di φ' .
Si ripongono β , $\emptyset_c$, $\emptyset_M$
Si dispone α
Si riscrive la Marca

M 0 - Conta +1 Cp
- Rip UMZ se nel preced. pdc è uscita una cifra significat.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disp $\emptyset_c$

M 6 - Sonda Mem
- Sonda Acc
- Disp. $\emptyset_M$
- Esce la Marca

F01 - GMUA

M 7 - GUAA
- Compl. Mem se D è disp.
- +1 Mem se D e UMZ sono disp.
- Rip IZ se da UA è uscita una cifra signif.

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt
- Disp φ'

M 8 - Poni RC in R
- Riscrivi Marca

M 8,5 - Riponi $\emptyset_M$, β , $\emptyset_c$

M 9 - Disp. α
- Sel T

M 9,5 - Rip φ' .

5.10.3.2.2. Sottocaso 2 B - Periodo prefinale -

Come per il pdc 3, inoltre:
La comp. dà esito favorevole, nasce $\emptyset_c$, indi $\emptyset_M$ e ad M6 esce $\emptyset_A$. Le condiz. R e D non consentono la nascita di φ' .
Nasce $\emptyset_A$.

M 0 - Conta +1 Cp
- Rip UMZ se nel preced. pdc è uscita una cifra sign.

M 3 - Conta -1 W
- Conta +1 V

M 5,5 - Compara
- Disp $\emptyset_c$

M 6 - Sonda Mem
- Sonda Acc
- Disp $\emptyset_M$
- Esce la Marca

F01 - GMUA

M 7 - GUAA
- Compl. Mem : se D è disp.
- +1 Mem se D e UMZ sono disp.
- Rip IZ se da UA è uscita una cifra significativa

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

M 8,5 - Disp $\emptyset_A$

Periodo finale

Nasce $\emptyset_{M1}$. Non si pilota Mem. Si aziona il gen. di zeri sul canale da Mem. complementando event. a 9 .
Si pilota Acc. senza sonda. Si aziona il gen. di zeri sul can. di Acc., complementando event. a 9 .
Si pone l'uscita dell'UA nella success.posiz. di Acc.
Nasce φ'. Si ripongono β , $\emptyset_C$, $\emptyset_A$, $\emptyset_M$, $\emptyset_{M1}$.
Si disp. α . Si sel. T .

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 3,5 - Disp. $\emptyset_{M1}$

M 4,5 - Pil Acc Int

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)

F01 - Zero Mem
- Zero Acc
- GMUA

M 7 - GUAA
- Compl. Mem se D è disp.
- Compl. Acc. se NC è in C
- Riponi IZ : se da UA è uscita una cifra signif.

M 7,5 - Pil Acc Scritt
- GMM
- Disp. φ'
- Scrivi Marca

M 8 - Poni RC in R

M 8,5 - Riponi β , $\emptyset_C$, $\emptyset_A$, $\emptyset_M$, $\emptyset_{M1}$

M 9 - Disp. α
- Sel T .

M 9,5 - Rip. φ' .

5.10.3.3. Procedura di fine tipo 3

Il numero in Acc. è di lunghezza maggiore del numero in Mem.

---

1) - E' superfluo.

5.10.3.3.1. Periodo n

Come per il periodo 3 ed inoltre:
La comparazione dà esito favorevole: nasce $\emptyset_C$, indi $\emptyset_M$.
Non si può avere fine perchè non è nato f_A .

M 0 - Conta +1 Cp
- Rip UMZ : se nel pdc preced. è uscita una cifra sign.

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int

M 5,5 - Compara
- Disp $\emptyset_C$

M 6 - Disp $\emptyset_M$
- Sonda Mem
- Sonda Acc

F01 - GMUA

M 7 - GUAA
- Compl.Mem. se D è disposto
- +1 Mem : se D e UMZ sono disposti
- Riponi IZ : se da UA è uscita una cifra significativa

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

5.10.3.3.2. Periodo n+1

Nasce $\emptyset_{M1}$, non si pilota Mem. Si aziona il gen. di zeri sul canale di Mem. Eventualmente si complementano a 9 gli zeri. Si estrae la success. cifra da Acc. e la si immette in UA 2 . Si pone l'uscita di UA nella succ.posiz. di Acc. Non essendo comparso f_A non nasce .

M 0 - Conta +1 Cp
- Rip UMZ : se nel prec. pdc è uscita una cifra sign.

M 3 - Conta -1 W
- Conta +1 V

M 3,5 - Disp. $\emptyset_{M1}$

M 4,5 - Pil Acc Int

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc

F01 - Zero Mem
- GMUA

M 7 - GUAA
- Compl. Mem. se D è disp.
- +1 Mem. : se D e UMZ sono disp.
- Rip. IZ : se da UA è uscita una cifra sign.

M 7,5 - Pil Acc Scritt
- GMM

I successivi periodi sono tutti eguali a n+1 fino a che non si presenta uno dei sottocasi seguenti.

5.10.3.3.3. Sottocaso 3 A

Periodo finale come per il p.d.c. n+1 inoltre:
Compare la Marca, R e D consentono la nascita di φ'. Si ripongono β , $\emptyset_c$, $\emptyset_M$, $\emptyset_{M1}$
Si dispone α , si sel T

M 0 - Conta +1 Cp
- Rip UMZ, se nel prec. pdc è uscita una cifra sign.

M 3 - Conta -1 W
- Conta +1 V

M 3,5 - Disp. $\emptyset_{M1}$ (1)

M 4,5 - Pil Acc Int

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)
- Sonda Acc
- Esce Marca

F01 - Zero Mem
- GMUA

M 7 - GUAA
- Compl. Mem se D è disp.
- Rip IZ se da UA è uscita cifra significativa.

---

1) - E' superfluo.

M 7,5 - Pil Acc Scritt
- GMM (sup.)
- Disp φ'

M 8 - Poni RC in R
- Rigenera Marca

M 8,5 - Rip. β , $\emptyset_c$, $\emptyset_M$, $\emptyset_{M'}$

M 9 - Disp. α
- Sel T

M 9,5 - Rip. φ'.

5.10.3.3.4. Sottocaso 3B

Periodo prefinale

Come per il pdc n+1 inoltre: compara la Marca che indica la fine del numero in Acc però nè R e D, nè CN e D, consentono la nascita di φ'.

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 3,5 - Disp. $\emptyset_{M1}$ (1)

M 4,5 - Pil Acc Int

M 5,5 - Compara (1)

M 6 - Sonda Mem
- Sonda Acc
- Esce Marca

F 01 - Zero Mem
- GMUA

M 7 - GUAA
- Compl.Mem : se D è disposto
- Rip IZ se da UA è uscita cifra significativa

M 7,5 - Pil Acc. Scritt.
- GMM (1)

M 8,5 - Compare $\emptyset_A$

periodo finale

E' nato $\emptyset_A$.

---

1) - E' superfluo.

Non si pilota Mem. Si aziona il generatore di zeri sul canale di Memoria, complementando eventualmente a 9 gli zeri.
Si pilota l'acc. senza sondare. Si aziona il generatore di zeri sul canale di Acc. complementando aventualmente a 9 .
Si pone l'uscita dell'UA nella successiva posizione di Acc. Si scrive la Marca.
Nasce φ'. Si ripongono β , $\emptyset_C$, $\emptyset_A$, $\emptyset_M$, $\emptyset_{MI}$.
Si dispone α , si sel T .
Comandi come nel sottocaso 2B .

5.11. ISTRUZIONE -MA

Sottrai Memoria da Accumulatore, ivi

Codice decimale : 27
Codice binario : 0111 1111
Decodif. parziali: Δ2 ΔA Δm
Configurazione : LdLuImIcIdIuTaFdFu

	0111	1111
Δ2	01..	.111
ΔA	..11	....
Δm	....	1...

←-1 W

Sottrae un numero di lunghezza LdLu (Lu modificato secondo Ta) proveniente da Memoria a partire dalla posizione ImIcIdIu (modificate secondo Ta), da un numero in Acc, a partire dalla posizione specificata dal contenuto del Raus P, di lunghezza fissata dalla Marca.

Per le altre caratteristiche dell'istruzione, vale quanto detto a proposito della +MA . Vedi fig. 5.11.1. e 5.11.2.

5.11.1. Preparazione ed esecuzione -MA

Come per la +MA .

5.12. ISTRUZIONE +MT

Addiziona Memoria a RT, ivi

Codice decimale : 32
Codice binario : 0110 0111
Decod. parziali : $\Delta 2$ ΔT Δp
Configurazione : LuTbImIcIdIuTaFdFu

	0110	0111
$\Delta 2$	01..	.111
ΔT	..10	....
Δp	....	0...

$\xleftarrow{\quad -1 \ W \quad}$

Somma un numero di lunghezza Lu proveniente da Memoria, a partire dalla posizione ImIcIdIu (modificata secondo Ta) con un numero nel RT, a partire dalla posizione Tb (modificata secondo Ta).
Sia l'istruzione che il contenuto del registro Ta vengono riscritti nelle loro rispettive posizioni. Il numero in Memoria viene riscritto nella sua posizione originale.
Il risultato viene scritto nel posto occupato dal numero del RT .
L'operazione può avvenire correttamente solo tra numeri non segnati. Di conseguenza non si aziona mai alcuno dei complementatori.
La fine dell'operazione è ottenuta con le stesse modalità che governano la fine dell'operazione di somma in Acc, ed inoltre, una volta comparsa la Marca, l'operazione termina nel pdc in cui compare $\emptyset_c$. L'eventuale riporto non viene scritto in RT, ma aziona l'indicatore di trabocco in T , OT . Per quanto riguarda la Marca, se essa non viene letta nel corso dell'operazione, non viene spostata. Se è stata letta, viene scritta nel pdc in cui compare il segnale φ'.

5.12.1. Preparazione +MT

Come per la preparazione MT .

5.12.2. Esecuzione +MT

5.12.2.1. Periodo 1

Si porta l'indirizzo di partenza da Raus 0 a W (2g).
Si completa l'indirizzo di partenza su V, contando +1 . Si estrae la prima cifra da Mem e la si immette in UA_1 (1).

---

1) - Nessuno dei complementatori viene mai usato in questa istruzione.-

Si estrae la prima cifra da RT e la si immette in UA_2 .
Si scrive l'uscita da UA nella prima posizione di RT .
Si confronta Cp con RL .

M 0 - Rim a 01 Cp
- Canc W
- Sel Raus P (1)
- Rip UMZ (1)

M 1 - Disp UMZ (1)

M 1,5 - Int Raus ⟶ W : è selezionato 0 da p3 MO

M 2 - Zero RL_{dec}

M 3 - Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc
- RC in C

F01 - GMUA

M 7 - Scritt W ⟶ Raus (2g)
- GUAA

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

5.12.2.2. Periodo 2

Si estrae la successiva cifra da Mem e la si immette in UA_1 .
Si estrae la successiva cifra da RT e la si immette in UA_2 .
Si scrive l'uscita da UA nella prima posizione di RT .
Si confronta Cp con RL .

M 0 - Conta +1 Cp
- Rip UMZ (1)

M 3 - Conta -1 W
- Conta +1 V

---

1) - E' superfluo.-

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc

FO1 - GMUA

M 7 - GUAA

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

I successivi periodi sono tutti uguali a p2 fino al periodo in cui ha inizio la procedura di fine.

5.12.2.3. Procedura di fine +MT

Come per la +MA , con le seguenti differenze :

- è selezionato T anzichè A
- non si complementa mai, nè da Mem, nè da RT .
- ci si riferisce al periodo tipo 2 anzichè al 3 .
- comparsa la Marca, l'operazione termina comunque nel pdc in cui compare $\emptyset_c$: l'eventuale riporto non viene scritto in RT , ma aziona OT .

5.13. ISTRUZIONE -MT

Sottrai Memoria da RT, ivi

Codice decimale: 37
codice binario : 0110 1111
Decod. Parziali: Δ 2 Δ T Δ m
Configurazione : LuTbImIcIdIuTaFdFu

	0110	1111
Δ 2	01..	.111
Δ T	..10	....
Δ m	....	1...

←—1 W

Sottrae un numero, di lunghezza Lu, proveniente da Mem a partire dalla posizione ImIcIdIu (modificata secondo Ta), da un numero nel RT, a partire dalla posizione Tb (modificata secondo Ta).
Le altre caratteristiche dell'istruzione sono identiche alle corrispondenti della +MT, con la sola differenza che il numero da memoria è complementato a 10 fino alla prima cifra significativa inclusa, e a 9 per le altre cifre.

5.13.1. Preparazione -MT

Come per la preparazione +MT .

5.13.2. Esecuzione -MT

5.13.2.1. Periodo 1

Come per il pdc 1 dello +MT ed inoltre: si complementa a 10 il numero da Mem.

M 0 - Rim a 01 Cp
- Canc W
- Sel Raus P (1)
- Rip UMZ (1)

M 1 - Disp UMZ

---

1) - E' superfluo.

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 MO

M 2 - Zero RL_{dec}

M 3 - Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di α

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc
- RC in C

F01 - GMUA

M 7 - Scritt W ⟶ Raus (2g)
- GUAA
- Compl Mem
- +1 Mem

M 7,5 - Pil Mem Scritt
- GMM
- Pil Acc Scritt

5.13.2.2. Periodo 2

Come per il pdc 4 della +MT ed inoltre: si complementa l'uscita da Mem a 10 se nel pdc precedente era uscito zero, a 9 se era uscita una cifra significativa.

M 0 - Conta +1 Cp
- Rip UMZ : se da Mem è uscita una cifra significativa nel pdc precedente

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è sel T

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc
- RC in C

F01 - GMUA

M 7 - GUAA
- Compl Mem
- +1 Mem : se UMZ è disposto

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt

I successivi periodi sono tutti uguali a p2 fino al pdc in cui ha inizio la procedura di fine.

5.13.2.3. Procedura di fine -MT

Come per la +MA, con le seguenti differenze:

- è selezionato T anzichè A
- si complementa sempre l'uscita da Mem
- ci si riferisce al pdc—tipo n°2 anzichè al n° 3
- comparsa la Marca, l'operazione termina appena compare $\emptyset_c$: l'eventuale riporto non viene scritto in RT, ma aziona OT .

5.14. ISTRUZIONE +IT

Somma istruzione con RT

Codice decimale : 42
Codice binario : 0100 0111
Decod.parziali : Δ 2 ΔI Δp
Configurazione : XLImIcIdIuTaFdFu

	0100	0111
Δ 2	01..	.111
Δ I	..00	....
Δ p	....	0...

← -1 W

Somma il numero LImIcIdIu dell'istruzione in Memoria, con il contenuto del registro Ta del RT . Il riporto fra Im ed L è impedito, dovendo l'istruzione servire esclusivamente per operare su modificatori. I caratteri dell'istruzione vengono riscritti in memoria nelle loro posizioni originali. Il risultato viene scritto nel registro Ta . L'operazione può avvenire correttamente solo fra numeri non segnati. Di conseguenza non si complementa mai, nè da Mem nè da Acc.
L'istruzione non è modificabile, ed è priva di fase esecutiva.
La somma si arresta al periodo p8 incluso: l'eventuale riporto non viene utilizzato e aziona l'indicatore di trabocco in T . Il carattere X serve solo a completare le strutture di 9 pdc dell'istruzione.

Ad ogni pdc da 4 a 8 inclusi si esamina l'uscita di UA : se è diversa da zero si ripone il flip-flop IZ . Al termine dell'operazione se IZ è ancora disposto, il risultato è nullo.

5.14.1. Preparazione ed esecuzione +IT

Periodi 1 e 2 come per la preparazione MA
Periodo 3 come per la preparazione ed esecuzione IT .

5.14.1.1. Periodo 4

Si pone in Vd il carattere estratto da Mem nel pdc precedente.
Si estrae la cifra Iu da Mem e la si immette in UA_1 .
Si estrae la prima cifra dal registro Ta e la si immette in UA_2 .
Si scrive l'uscita da UA nella prima posizione di Ta (1).

---

1) - L'uscita di UA viene immessa anche, inutilmente, in Raus 0 unità.-

M 0 - Conta +1 Cp
- Canc V : vedi note

M 1,5 - Canc Raus M (è selezionato 0 da MO p3)

M 3 - Conta -1 W
- UA ⟶ Vd : pone in Vd il carattere Ta uscito da Mem nel pdc precedente. Vedi nota a pag.71
- Conta +1 V : Vu passa a zero codice
- d ⟶ Vu : vedi p 4 pag.71

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

M 7 - GUAA
- GUAG (1)
- UA ⟶ Raus M (1)
- Riponi IZ : solo se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM

5.14.1.2. Periodo 5

Si estrae la cifra Ld da Mem e la si immette in UA_1 .
Si estrae la seconda cifra dal registro Ta e la si immette in UA_2 .
Si scrive l'uscita da UA nella seconda posizione di Ta (2).

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

---

1) - E' superfluo.-
2) - L'uscita da UA viene anche scritta, inutilmente, in Raus 0 decine.-

M 7 - GUAA
- GUAG (1)
- UA ⟶ Raus M (1)
- Riponi IZ : solo se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM

5.14.1.3. Periodo 6

Come per il periodo 5, ma riferito alle centinaia di I ed alla 3^ posizione di Ta .

5.14.1.4. Periodo 7

Vedi fig. 5.14.1.
Come per il pdc 5, ma riferito alle **centinaia di** I ed alla 4^ posizione di Ta . Il riporto decimale è **sospeso.** Nasce un segnale, φ , che serve solo nella CIT .

5.14.1.5. Periodo 8

Vedi fig. 5.14.2.
Come per il pdc 5, ma riferito a L ed alla 5^ posizione di Ta (2).

5.14.1.6. Periodo 9

Come per il pdc 9 della IT .

---

1) - E' superfluo.
2) - L'uscita da UA viene immessa anche in $RL_{un.}$ ma ciò è superfluo.-

5.15. ISTRUZIONE -IT

Sottrae l'istruzione dal RT

Codice decimale : 47
Codice binario : 0100 1111
Decod. parziali : Δ 2 ΔI Δm
Configurazione : XLImIcIdIuTaFdFu

	0100	1111
Δ 2	01..	.111
Δ I	..00	....
Δ m	....	1...

← - 1 W

Sottrae il numero LImIcIdIu dell'istruzione in Mem., dal contenuto del registro Ta del RT . Le altre caratteristiche sono identiche alla +IT, salvo che si complementa l'operando da Mem (a 10 fino alla prima cifra significativa inclusa, a 9 per le altre cifre) (1).

5.15.1. Preparazione ed esecuzione -IT

Periodi 1 e 2 come per la preparazione MA .
Periodo 3 come per la prep. ed esec. IT ed inoltre:
a M 8 si dispone D' .

5.15.1.1. Periodo 4

Si pone in Vd il carattere Ta estratto da Mem nel pdc precedente.
Si estrae la prima cifra Iu da Mem e la si immette in UA_1 , complementandola a 10 .
Si estrae la prima cifra dal registro Ta e la si immette in UA_2 .
Si scrive l'uscita da UA nella prima posizione di Ta (2)

M 0 - Conta +1 Cp
- Canc. V : vedi nota (2) pag. 71 .

M 1,5 - Canc. Raus M (è selezionato 0 da MO p3) (3)

M 3 - Conta -1 W
- UA ⟶ Vd : pone in Vd il carattere Ta uscito da Mem nel pdc precedente. Vedi nota a pag. 71 .

---

1) - La decisione se si deve complementare a 10 o a 9 è presa in base all'esistenza del riporto decimale, anzichè in base allo stato di flip-flop UMZ .
2) - L'uscita da UA viene immessa anche, inutilmente, in Raus 0 un.
3) - E' superfluo.-

M 3 - Conta +1 V : Vu passa a zero codice
- d ⟶ Vu : vedi pag.71 pdc 4 .
- Disponi +1 Mem.

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

M 7 - GUAA
- GUAG (1)
- UA ⟶ Raus M (1)
- Compl Mem
- + 1 Mem
- Riponi IZ : solo se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM

5.15.1.2. Periodo 5

Vedi fig. 5.15.1.
Si estrae la cifra Id da Mem e la si immette in UA_1 complementandola a 10 se nel pdc precedente si è avuto riporto, a 9 se non si è avuto.
Si estrae la seconda cifra da Ta e la si immette in UA_2 . Si scrive l'uscita da UA nella seconda posizione di Ta (2).

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int (è selezionato T dall'inizio di α)

M 6 - Sonda Mem
- Sonda Acc

F01 - GMUA

---

1) - E' superfluo.-
2) - L'uscita da UA viene immessa anche, inutilmente, in Raus 0 decine.-

M 7 - GUAA
- GUAG (1)
- UA ⟶ Raus M (1)
- Compl Mem
- +1 Mem : se nel precedente pdc c'è stato riporto.
- Riponi UA : se da UA è uscita una cifra significativa.

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt
- GMM

5.15.1.3. Periodo 6

Come per il pdc 5, ma riferito alle centinaia di I e alla 3^ posizione di Ta .

5.15.1.4. Periodo 7

Come per il pdc 5, ma riferito alle migliaia di I e alla 4^ posizione di Ta . E' sospeso il riporto decimale. Nasce un segnale che serve solo nella CIT .

5.15.1.5. Periodo 8

Come per il pdc 5, ma riferito a L ed alla 5^ posizione di Ta (2).

5.15.1.6. Periodo 9

Come per il pdc 9 della IT .

---

1) - E' superfluo.-
2) - L'uscita da UA viene anche scritta, inutilmente, in $RL_{un.}$.-

5.16. ISTRUZIONE ITT

Sottrae Tb da I (mod Ta). Risultato in Tb

Codice decimale : 34
Codice binario : 0110 0100
Decod. parziali : Δ 4 Δ T Δ p
Configurazione : XTbImIcIdIuTaFdFu

	0110	0100
Δ 4	01..	.100
Δ T	..10	....
Δ p	....	0...

← - 1 W

Sottrae il contenuto delle prime quattro posizioni del registro Tb di RT (modificato secondo Ta) dai caratteri ImIcIdIu dell'istruzione (modificati secondo Ta). Il risultato occupa le prime quattro posizioni del registro Tb (mod Ta).
I caratteri dell'istruzione vengono riscritti in Mem nelle loro posizioni originali.
L'operazione può avvenire correttamente solo fra numeri non segnati.
Il carattere X serve solo a completare la struttura di 9 pdc della istruzione.
L'esecuzione dura sempre e solo quattro pdc, durante i quali si complementa il numero uscente dal Modificatore (a 10 fino alla prima cifra significativa inclusa, a 9 per le altre).

5.16.1. Preparazione ITT

Come per la MT .

5.16.2. Esecuzione ITT

5.16.2.1. Periodo 1

Vedi fig. 5.16.1.
Si porta il numero ImIcIdIu (modificato secondo Ta) in parallelo da Raus 0 a W (2g).
Si completa l'indirizzo di partenza su Vu, contando +1 .
Senza pilotare Mem, si estrae il contenuto di Wu e lo si immette (attraverso lo stat. d'uscita Mem) in UA_1 .
Si estrae la prima cifra dal RT e la si immette da UA_2, complementandola a 10 .
Si scrive l'uscita da UA nella prima posizione di Mod.

M 0 - Rim a 01 Cp
- Canc W
- Sel Raus P (1)
- Rip UMZ (1)

M 1 - Disp UMZ (1)
- Disp UAZ

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 MO

M 3 - Conta +1 V

M 4,5 - Pil Acc Int : è selezionato T dall'inizio di α .

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)
- Sonda Acc
- RC in C
- Wu ⟶ Stat Mem

F01 - GMUA

M 7 - Scritt W ⟶ Raus (2g)
- GUAA
- Compl Acc
- +1 Acc

M 7,5 - Pil Acc Scritt
- GMM (1)

M 9 - Rip UAZ : se è uscita da RT una cifra significativa).

5.16.2.2. Periodo 2

Vedi fig. 5.16.2.
Senza pilotare Mem (nè contare -1 W), si estrae il contenuto di Wd e lo si immette (attraverso lo stat d'uscita Mem) in UA_1 .
Si estrae la seconda cifra dal RT e la si immette in UA_2 , complementandola a 10 se nel pdc 1 è uscito zero, a 9 se è uscita una cifra significativa.
Si scrive l'uscita da UA nella seconda posizione di RT .

M 0 - Conta +1 Cp
- Rip UMZ (1).

---

1) - E' superfluo.-

M 3 - Conta +1 V

M 4,5 - Pil Acc Int (è selezionato T dall'inizio di α).

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)
- Sonda Acc
- Wd $\longrightarrow$ Stat Mem

F01 - GMUA

M 7 - GUAA
- Compl Acc
- +1 Acc se UAZ è riposto

M 7,5 - Pil Acc Scritt
- GMM (1)

M 9 - Rip UAZ : se è uscita da Mod una cifra significativa.

5.16.2.3. Periodo 3

Come per il pdc 2, ma riferito alle centinaia di W ed alla terza posizione di RT .

5.16.2.4. Periodo 4

Come per il pdc 2, ma riferito alle migliaia di W e alla quarta posizione di RT . Inoltre: nasce φ , si ripone β , si dispone α.

---

1) - E' superfluo.-

5.17. ISTRUZIONE Y

Da Memoria a TO

Codice decimale : 24
Codice binario : 0111 0100
Decod. parziali : Δ 4 ΔA Δp (1)
Configurazione : LdLuImIcIdIuTaFdFu

	0111	0100
Δ 4	01..	.100
Δ A	..11	....
Δ p	....	0...

← - 1 W

Trasferisce (attraverso apposito canale esterno alla tavola di somma di UA) una parola di lunghezza LdLu (Lu modificata secondo Ta), dalla Memoria, a partire dalla posizione ImIcIdIu (modificata secondo Ta), al RT, a partire dalla posizione fissa 0 .
Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali.
La parola trasferita è riscritta in Mem nella sua posizione originale.
Se la parola in Mem è segnata, il segno non entra in RT, ma va a disporre (se è -) il flip-flop SR (2).
La parola può occupare al massimo l'intero registro TO . Pertanto LdLu può essere al massimo 10 senza segno, 11 con segno.
Una marca viene scritta nell'ultima posizione raggiunta.

5.17.1. Preparazione Y

Come per la preparazione MA ed inoltre al pdc 3 non si azzerano gli indicatori di Acc., in p3 M3,5 si ripone SR .

5.17.2.1. Periodo 1

Vedi fig. 5.17.1.
Si porta l'indirizzo dell'operando da Raus 0 a W (2g).
Si seleziona TO su V, scrivendo zero sulle decine e contando +1 .

---

1) - Si noti che la Y ha una dec. parz. Δ A pur essendo una operazione su T : ciò consente di non introdurre il segno in T.
2) - Se il segno è + , va perduto, in quanto non serve nel prodotto dei segni.

Si estrae il primo carattere da Mem e lo si pone nella prima posizione di TO, attraverso il canale esterno alla tavola di somma.
Se è un segno meno, si dispone il flip flop SR
Si confronta Cp con RL .

M 0 - - Rim a 01 Cp
- Canc W
- Sel Raus P (1)

M 1,5 - Zero dec.

M 3 - Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di pur essendo una istruzione tipo Δ A .

M 5,5 - Compara (1g)

M 6 - Sonda Mem
- Sonda Acc (1)
- RC in C

F01 - GMUA (1)

M 7 - GMA
- Scritt W ⟶ Raus (1)
- GMM
- Pil Acc Scritt
- GAA : solo se da Mem è uscito un segno

M 8,5 - Disp. SR : se da Mem è uscito un segno meno.

5.17.2.2. Periodo 2

Si estrae la seconda cifra da Mem. Se la prima fu un segno, si scrive questa seconda nella prima posizione di TO, (cancellando quanto scrittovi nel pdc 1).
Se la prima non fu un segno, si scrive la seconda cifra uscente da Mem nella seconda posizione di TO, sempre attraverso il canale esterno alla tavola di somma.
Si confronta Cp con RL .

---

1) - E' superfluo.-

M 0 - Conta +1 Cp

M 3 - Conta -1 W
- Conta +1 V : solo se nello Stat d'uscita Mem non c'è segno.

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di α

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc (1)

F01 - GMUA

M 7 - GMA

M7,5 - Pil Mem Scritt
- Pil Acc Scritt

I successivi periodi sono uguali ai corrispondenti della MA, con l'unica differenza che è selezionato T anzichè A .

---

1) - E' superfluo.-

5.18. ISTRUZIONE DA

Disponi Accumulatore

Codice decimale : 39
Codice binario : 0110 1100
Decod. Parziali : Δ4 ΔT Δm
Configurazione : XXXXIdIuTaFdFu

	0110	1100
Δ 4	01..	.100
ΔT	..10	....
Δm	....	1...

← - 1 W

Trasferisce i caratteri IdIu dell'istruzione, modificati secondo Ta, dalla Memoria al Raus P . E' priva di fase escutiva e si esaurisce in cinque pdc : i quattro caratteri X servono solo a completare i 9 pdc di preparazione.
Sia l'istruzione, che il contenuto del registro Ta vengono riscritti nelle loro posizioni originali.

5.18.1. Preparazione ed esecuzione DA

Periodi 1, 2 come per la prep. MA .

5.18.1.1. Periodo 3

Si cancella (interrogandolo il Raus P . Si estrae da Mem il carattere Ta che (sommato +0), attraversa l'UA ed entra nello stat di governo.

M 0 - Conta +1 Cp
- Sel Raus P

M 1,5 - Int Raus → V : serve a cancellare Raus P

M 3 - Conta -1 W

M 4,5 - Pil Mem Int

M 6 - Sonda Mem

F01 - GMUA
- Zero Acc

M 7,5 - Pil Mem Scritt
- GMM

M 8,5 - Rip S (1)
- Rip -/+ (1)
- Rip CN (1)

Periodi 4 e 5 come i corrispondenti della prep. MA . Inoltre l'uscita da UA viene scritta in Raus P, rispettivamente unità e decine, vedi fig. 5.18.1.

La scrittura in Raus M è superflua.

I periodi 6, 7, 8 sono superflui: tutto avviene come per i corrispondenti pdc della MA (preparazione).

Periodo 9 : come per il pdc 9 della IT .

---

1) - E' superfluo.-

5.19. ISTRUZIONE FAM

Lunghezza di Acc in Mem

Codice decimale : 2
Codice binario : 0111 0000
Decod. parziali : $\Delta\omega$ ΔA Δp
Configurazione : XXImIcIdIuTaFdFu

	0111	0000
$\Delta\omega$	01..	.000
ΔA	..11	....
Δp	....	0...

$\xleftarrow{\quad}$ -1 W

Scrive in Memoria, all'indirizzo ImIcIdIu (modificato secondo Ta) il numero indicante la lunghezza effettiva (non quella stabilita dalla Marca) del numero contenuto in Acc., a partire dalla posizione specificata nel contenuto di Raus P .

Sia l'istruzione, che il contenuto di Ta vengono riscritti nelle loro posizioni originali.

I caratteri X sono superflui e servono solo a completare i 9 pdc della preparazione.

L'esecuzione consta di due sottofasi: nella prima si legge l'Acc: ad ogni pdc si copia il contenuto del cont p sul RL (previa cancellazione) se l'uscita da Acc è significativa, si lascia inalterato RL se l'uscita da Acc è zero. Nella seconda (che dura solo 2 pdc) si trascrive in Mem, all'indirizzo voluto dall'istruzione , il contenuto di RL, che rappresenta la lunghezza effettiva del numero in Acc.

5.19.1. Preparazione FAM

Come la prep. MA, tenendo conto che gli ultimi due pdc sono superflui. Gli indicatori di Acc non vengono cancellati al pdc 3 .

5.19.2. Esecuzione FAM

5.19.2.1. Periodo 1 della fase β

Vedi fig. 5.19.1.
Si porta l'indirizzo di Mem a cui si dovrà scrivere la lunghezza cercata, dal Raus O a W (2 g). Si porta l'indirizzo di partenza dell'Acc da Raus P a V e lo si copia poi di nuovo indietro in Raus P .

Si pilota Mem senza Sonda.
Si scrive zero codice su RL (1)
Si estrae la prima cifra da Acc e la si immette in UA_1 . Si immette zero codice in UA_2 : se l'uscita di UA è una cifra significativa si cancella RL e vi si copia il contenuto di Cp .

M 0 - Rim a 01 Cp
- Canc W
- Canc V
- Sel Raus P

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 MO
- Int Raus ⟶ V : è selezionato P da MO

M 2 - Zero RL

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato A dall'inizio di β

M 5 - Scritt V ⟶ Raus A : è selezionato P da MO

M 5,5 - Compara (1 g) (2)

M 6 - Sonda Acc
- RC in C

F01 - GMUA
- Zero Mem

M 6,5 - Canc RL : se da UA è uscita una cifra significativa

M 7 - Scritt W ⟶ Raus (2)

M 7,5 - Pil Mem Scritt (2)
- GMM (1)
- Pil Acc Scritt
- GAA

M 8 - Cp ⟶ RL : se da UA è uscita una cifra significativa.

5.19.2.2. Periodo 2 della fase

Si pilota ancora Mem senza sonda sullo stesso indirizzo. Si estrae la successiva cifra da Acc e la si immette in UA_1 . Si immette zero codice in UA_2 . Se l'uscita da UA è una cifra significativa

---

1) - In tal modo, se da Acc escono sempre zeri fino alla Marca (inclusa), su RL resta zero codice.-
2) - E' superfluo.-

si cancella RL e vi si copia Cp .

M 3 - Conta +1 V

M 4,5 - Pil Mem Int (1)
- Pil Acc Int

M 5,5 - Compara (1)

M 6 - Sonda Acc

F01 - GMUA
- Zero Mem

M 6,5 - Canc RL : se da UA è uscita una cifra significativa

M 7,5 - Pil Mem Scritt (1)
- GMM (1)
- Pil Acc Scritt
- GAA

M 8 - Cp $\longrightarrow$ RL : se da UA è uscita una cifra significativa.

I successivi periodi sono tutti uguali a p2, fino al pdc finale in cui esce la Marca.

5.19.2.3. Periodo finale della fase β

Come per il pdc 2 ed inoltre.

M 6 - Esce la Marca

M 7,5 - Disponi φ'
- Riscrivi Marca
- RC in R

M 8 - Riponi β

M 9 - Disponi γ

5.19.2.4. Periodo 1 della fase γ

Fig. 5.19.2.
Si scrive il contenuto di RL unità nella posizione di Memoria

---

1) - E' superfluo.-

ImIcIdIu (mod Ta) sempre presente su W .
Si legge e si riscrive (inutilmente) la successiva posizione di Acc.

M 3 - Conta +1 V (1)

M 4,5 - Pil Mem Int
- Pil Acc Int (1)

M 5,5 - Compara

M 6 - Sonda (1)
- RC in C

FO1 - GMUA (1)
- Zero Mem (1)

M 7 - $RL_{un.}$ ⟶ Mem

M 7,5 - Pil Mem Scritt
- Pil Acc Scritt (1)
- GAA (1)

5.19.2.5. Periodo 2 della fase γ

Si scrive il contenuto di RL decine nella successiva posizione di Mem.
Si legge e si riscrive (inutilmente) la successiva posizione di Accumulatore.
Nasce φ . Ripone β , γ , RC , φ' . Dispone α

M 0 - Conta +1 Cp
- Nasce

M 3 - Conta -1 W
- Conta +1 V (1)

M 4,5 - Pil Mem Int
- Pil Acc Int (1)

M 5,5 - Compara

M 6 - Sonda Acc (1)

FO1 - GMUA (1)
- Zero Mem (1)

M 7 - RL ⟶ Mem
M 8 - RC in R (grazie a Δ Q)
M 8,5 - Riponi γ
M 9 - Disp α
M 9,5 - Rip. φ'

---

1) - E' superfluo.-

5.20. ISTRUZIONE CMA

Confronta Memoria con Accumulatore

Codice decimale : 17
Codice binario : 0011 1111
Decod. parziali : ΔC ΔA Δm
Configurazione : LdLuImIcIdIuTaFdFu

	0011	1111
ΔC	.0..	1111
ΔA	..11	....
Δm	....	1...

←1 W

Confronta un numero, di Lunghezza LdLu (Lu modificato secondo Ta), esistente in Memoria, a partire dalla posizione ImIcIdIu (modificato secondo Ta), con un numero esistente in Acc, a partire dalla posizione specificata dal contenuto del Raus P, di lunghezza fissata dalla Marca.

Il risultato del confronto è espresso dallo stato dei flipflop IC (Mem ≥ Acc, Mem ≤ Acc) e IZ (Mem=Acc, Mem≠Acc).

Il confronto è ottenuto attraverso una normale sottrazione tipo -MA (Acc - Mem), che lascia però inalterato il contenuto iniziale di Acc.

Ad ogni pdc dell'esecuzione si esamina l'uscita da UA : se è diversa da zero si ripone il flip-flop IZ .

Al termine della sottrazione si esamina lo stato dei seguenti segnali: SA , D , R per definire lo stato di IC .

Sia l'istruzione che il contenuto di Ta vengono riscritti nelle loro posizioni originali.
Il numero in Mem può essere segnato o non segnato. L'Acc può essere segnato o non segnato: se è segnato, può essere normale o in complemento.
Sia il numero in Memoria, che il numero in Acc, nonchè gli indicatori di Acc non vengono alterati dall'operazione.

5.20.1. Preparazione CMA

Come per la preparazione MA, senza azzerare al pdc gli indicatori di Accumulatore.

5.20.2. Esecuzione CMA

Vedi fig. 5.20.1. e 5.20.2.
Come per l'esecuzione -MA, con le seguenti differenze.

M 7 di ogni pdc : GUAA non si apre mai
M 7 di ogni pdc : se da UA esce una cifra significativa, si ripone IZ .
A ogni pdc : GAA si apre sempre.

Nel pdc finale a M 8 si dispone IC , se le condizioni degli indicatori di Acc ed il riporto lo consentono.

5.21. ISTRUZIONE CMT

Confronta Memoria con RT

Codice decimale :&7
Codice binario : 0100 1111
Decod. parziali : Δ C ΔT Δ m
Configurazione : LTbImIcIdIuTaFdFu

	0010	1111
Δ C	.0..	1111
Δ T	..10	....
Δ m	....	1...

← - 1 W

Confronta un numero di lunghezza Lu, esistente in Memoria, a partire dalla posizione ImIcIdIu (modificata secondo Ta) con un numero in RT, a partire dalla posizione Tb (modificata secondo Ta).

Il risultato del confronto è espresso dallo stato dei flip-flop IC (Mem ≥ T, Mem ≤ T) e IZ (Mem = T, Mem ≠ T).

Il confronto è ottenuto attraverso una normale sottrazione tipo -MT (Mod - Mem), che lascia però inalterato il contenuto iniziale di RT .

Ad ogni pdc si esamina l'uscita da UA : se è diversa da zero si ripone il flip-flop IZ .

Al termine della sottrazione si esamina lo stato dei segnali: D , R , per definire lo stato di IC .

Sia l'istruzione, che il contenuto di Ta vengono riscritti nelle loro posizioni originali.

L'operazione può avvenire correttamente solo fra numeri non segnati.
Sia il numero in Mem che quello in Mod non vengono alterati dall'operazione.

5.21.1. Preparazione CMT

Come per la preparazione MT .

5.21.2. Esecuzione CMT

Come per l'esecuzione -MT, con le seguenti differenze:
M 7 di ogni pdc : GUAA non si apre mai.
M 7 di ogni pdc : se da UA esce una cifra significativa, si rip.IZ
M 7,5 di ogni pdc : GAA si apre sempre.

Nel pdc finale a M 8 si dispone eventualmente IC .

5.22. ISTRUZIONE CIT

Confronta istruzione con RT

Codice decimale : =7
Codice binario : 0000 1111
Decod.parziali : Δ C Δ I Δ m
Configurazione : XLImIcIdIuTaFdFu

	0000	1111
Δ C	.0..	1111
Δ I	..00	....
Δ m	....	1...

← - 1 W

Confronta il numero ImIcIdIu dell'istruzione in Memoria, con il numero contenuto nel registro Ta di RT .

Il risultato del confronto è espresso dallo stato dei flip flop IC (Ist ⩾ T, Ist ⩽ T) e IZ (Ist = T, Ist ≠ T). Lo stato di IZ dipende anche dalla cifra L dell'istruzione.
Il confronto è ottenuto attraverso una normale sottrazione tipo -IT (T - I) che lascia però inalterato il contenuto iniziale del Mod. In ogni pdc da 4 a 8 si esamina l'uscita da UA : se è diversa da zero si ripone il flip flop IZ .

Al settimo pdc (quando è terminata la somma ImIcIdIu) si esamina lo stato dei segnali: D , R , per definire lo stato di IC .
I caratteri dell'istruzione vengono riscritti nelle loro posizioni originali.
L'operazione può avvenire correttamente solo fra numeri non segnati.
L'istruzione non è modificabile ed è priva di fase esecutiva.
Il carattere X serve solo a completare i 9 pdc di preparazione.

5.22.1. Preparazione ed esecuzione CIT

Come per la preparazione ed esecuzione -IT con le seguenti differenze:

M 7 dei pdc da 4 a 8 : GUAA non si apre.
M 7 dei pdc da 4 a 8 : se da UA esce una cifra significativa, si ripone IZ .
M7,5 dei pdc da 4 a 8 : GAA si apre.
M 8 del pdc 7 : è nato (a MO) φ , onde si procede a disporre eventualmente IC , vedi figura 5.22.1.

5.23. ISTRUZIONE SC >

Salta se (nell'ultimo confronto) Mem >

Codice decimale : 01
Codice binario : 0001 0011
Decod. parziali : Δ S Δ Q Δ p $b\bar{c}$
Configurazione : XTbImIcIdIuTaFdFu

← 1 W

	DCBA	dcba
	0001	0011
Δ S	0001	.
Δ Q	1	
Δ p		0
bc		.01.

Se nell'ultimo confronto la Mem è risultata maggiore, fa preparare alla macchina (anzichè la successiva istruzione) l'istruzione che si trova all'indirizzo ImIcIdIu modificato secondo Ta (1).
Se nell'ultimo confronto la Mem non è risultata maggiore, si prepara la successiva istruzione.
Ciò è ottenuto nel modo seguente.
Nei primi due pdc della preparazione si verifica la condizione (cioè lo stato di IC e IZ): se non è verificata, nel terzo pdc si arresta la preparazione della SC > e si passa a preparare la successiva istruzione.
Se la condizione è verificata, si sostituisce il contenuto del Raus I con l'indirizzo ImIcIdIu modificato secondo Ta.
In una successiva esecuzione di 4 pdc si scrive da W nel registro Tb del RT l'ultimo indirizzo (quello del carattere X) cui si è pervenuti nella preparazione di SC > (2).

5.23.1. Preparazione SC >

Periodi 1 e 2 come per la preparazione MA ed inoltre: al M8,5 di p2 compare il segnale CV ; se IC è riposto. Diversamente è presente il segnale $\overline{CV}$.

---

1) - Questo indirizzo corrisponde al carattere Fu dell'istruzione a cui si vuole saltare, in quanto la conta + 10 W è soppressa per le preparazioni che seguono un salto con condizione verificata.
2) - E' così possibile rientrare nel programma al punto interrotto dal salto attraverso un altro salto all'indirizzo contenuto in Tb, sommato + 17 .

5.23.1.1. Periodo 3 $\overline{CV}$

Si estrae (inutilmente) il carattere Ta, che (sommato + 0) attraversa UA ed entra nello stat di governo.

M 0 - Conta +1 Cp (1)

M 1,5 - Int Raus W (1)

M 3 - Conta -1 W (1)

M 4,5 - Pil Mem Int (1)

M 6 - Sonda Mem (1)

FO1 - GMUA (1)
- Zero Acc (1)

M 7,5 - Pil Mem Scritt (1)
- GMM (1)

I successivi periodi $\overline{CV}$ sono identici ai corrispondenti della preparazione MT fino al pdc 7 in cui compare il segnale φ che segna il termine dell'operazione.

5.23.1.2. Periodo 3 CV

Come per il periodo 3 della MA, senza azzerare gli indicatori di Acc.

5.23.1.3. Periodo 4 CV

Vedi fig. 5.23.1.
Come per il pdc 4 della MA, salvo che non si seleziona Raus 0 : pertanto la cifra Iu (modificata o no) viene immessa in Raus I (unità) selezionato da MO di α p1 .

5.23.1.4. Periodo 5 CV

Come per il pdc 5 della MA, ma sempre tenendo presente che è selezionato I anzichè 0 .

---

1) - E' superfluo.-

5.23.1.5. Periodo 6 CV

Come per il pdc 5 CV , ma riferito alle centinaia di I .

5.23.1.6. Periodo 7 CV

Vedi fig. 5.23.2.
Come per il pdc 5 CV, ma riferito alle migliaia di I .

5.23.1.7. Periodo 8 CV

Come per il pdc 8 della preparazione MT .

5.23.1.8. Periodo 9 CV

Come per il periodo 9 della preparazione MT, riferita però al carattere X (superfluo) anzichè al carattere Lu .

5.23.2. Esecuzione SC >

5.23.2.1. Periodo 1

Si completa l'indirizzo di partenza su Vu, contando +1 . Senza pilotare Mem, si estrae il contenuto di Wu e lo si pone in RT nella prima posizione del registro Tb, attraverso UA, sommando +0 . La sonda Acc è soppressa

M 0 - Rim a 01 Cp
- Canc V (1)
- Sel Raus P (1)

M 3 - Conta +1 V

M 4,5 - Pil Acc Int: è selezionato T dall'inizio di α .

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)
- RC in C
- Wu ⟶ Stat Mem

---

1) - E' superfluo.-

FO1 - GMUA
- Zero Acc

M 7,5 - Pil Acc Scritt
- GMM (1)

5.23.2.2. Periodo 2

Senza pilotare Mem (nè contare -1 W) si estrae il contenuto di Wd e lo si pone nella seconda posizione di Tb, attraverso UA, sommando +0 . La sonda Acc è soppressa.

M 0 - Conta +1 Cp

M 3 - Conta + 1 V

M 4,5 - Pil Acc Int : è selezionato T dall'inizio di α .

M 5,5 - Compara (1)

M 6 - Sonda Mem (1)
- Wd ⟶ Stat Mem

FO1 - GMUA

M 7,5 - Pil Acc Scritt
- GMM (1)

5.23.2.3. Periodo 3

Come per il pdc 2, ma riferito alle centinaia di W e alla terza posizione di RT .

5.23.2.4. Periodo 4

Come per il pdc 2, ma riferito alle migliaia di W e alla quarta posizione di RT . Inoltre nasce φ , si ripone β , si dispone α .

---

1) E' superfluo.

5.24. ISTRUZIONE SC <

Salta se nell'ultimo confronto Mem <

Codice decimale : 02
Codice binario : 0001 0111
Decod.parziali : ΔS ΔQ Δp ac
Configurazione : XTbImIcIdIuTaFdFu

← - 1 W

	0001	0111
Δ S	0001	
Δ Q	1	
Δ p		0
ac		.1.1

Come per la SC > , per la condizione che nell'ultimo confronto la Mem sia risultata minore di Acc (M ≤ A, IZ ≠ 0).

5.25. ISTRUZIONE SA >

Salta se Accumulatore > 0

Codice decimale : 03
Codice binario : 0001 0110
Decod. parziali : Δ S Δ Q Δp $\bar{a}b$
Configurazione : XTbImIcIdIuTaFdFu

	0001	0110
Δ S	0001	
Δ Q	1	
Δ p		0
$\bar{a}b$		.1.0

← - 1 W

Come per la SC > , per la condizione che Acc sia maggiore di zero (A ⩾ 0, (1), IZ ≠ 0), vedi fig. 5.25.1.

1) - A > 0 è ottenuto esaminando lo stato dei flip flop -/+ e CN .

5.26. ISTRUZIONE SA <

Salta se Accumulatore < 0

Codice decimale: 04
Codice binario : 0001 0100
Decod.parziali : Δ S ΔQ Δp $\overline{b}$c
Configurazione : XTbImIcIdIuTaFdFu

	0001	0100
Δ S	0001	
Δ Q	1	
Δ p		0
$\overline{b}$c		10

← - 1 W

Come per la SC > , per la condizione che Acc sia minore di zero (A < 0, (1), IZ ≠ 0).

1) - A < 0 è ottenuto esaminando lo stato dei flip flop -/+ e CN .

5.27. ISTRUZIONE S=

Salta se l'ultimo risultato aritmetico è stato zero o l'ultimo confronto ha dato uguaglianza.

Codice decimale : 05
Codice binario : 0001 1001
Decod. parziali : Δ S ΔQ Δm a$\bar{b}$
Configurazione : XTbImIcIdIuTaFdFu

	0001	1001
Δ S	0001	
Δ Q	1	
Δ m		1
a$\bar{b}$		01

← - 1 W

Come per la SC > , per la condizione che l'ultimo risultato aritmetico sia stato zero, oppure l'ultimo confronto abbia dato uguaglianza. (IZ = 0). Vale la condizione che si è verificata per ultima prima dell'istruzione S= .

5.28. ISTRUZIONE S≠

Salta se l'ultimo risultato non è stato zero o l'ultimo confronto non ha dato uguaglianza.

Codice decimale : 06
Codice binario : 0001 1011
Decod. parziali : $\Delta S\ \Delta Q\ \Delta m\ \ b\bar{c}$
Configurazione : XTbImIcIdIuTaFdFu

	0001	1011
ΔS	0001	
ΔQ	1	
Δm		1
$b\bar{c}$		01

← - 1 W

Come per la SC > , per la condizione che l'ultimo risultato aritmetico sia stato diverso da zero, o l'ultimo confronto non abbia dato uguaglianza (IZ ≠ 0).

Vale la condizione che si è verificata per ultima prima della istruzione S ≠ .

5.29. ISTRUZIONE STO

Salta se ci fu un trabocco dall'ultimo T operato

Codice decimale : 07
Codice binario : 0001 1111
Decod. parziali : ΔS ΔQ Δm ac
Configurazione : XTbImIcIdIuTaFdFu

	0001	1111
ΔS	0001	
ΔQ	1	
Δm		1
ac		.1.1

← - 1 W

Come per la SC > , per la condizione che l'ultima operazione aritmetica su RT abbia provocato un trabocco (OT disposto).

5.30. ISTRUZIONE SE 1

Salta sulla condizione esterna 1

Codice decimale : 08
Codice binario : 0001 1110
Decod. parziali : ΔS ΔQ Δm $\bar{a}$b
Configurazione : XTbImIcIdIuTaFdFu

	0001	1110
ΔS	0001	
ΔQ	1	
Δm		1
$\bar{a}$b		10

← - 1 W

Come per la SC > , per la condizione esterna 1 .

5.31. ISTRUZIONE SE 2

Salta sulla condizione esterna 2

Codice decimale : 09
Codice binario : 0001 1100
Decod. parziali : ΔS ΔQ Δm $\overline{b}$c
Configurazione : XTbImIcIdIuTaFdFu

	0001	1100
ΔS	0001	
ΔQ	1	
Δm		1
$\overline{b}$c		.10.

← - 1 W

Come per la SC > , per la condizione esterna 2 .

5.32. ISTRUZIONE SE 3

Salta sulla condizione esterna 3

Codice decimale : 0=
Codice binario : 0001 0000
Decod. parziali : Δ S Δ Q Δ p $\overline{abc}$
Configurazione : XTbImIcIdIuTaFdFu

	0001	0000
Δ S	0001	
Δ Q	1	
Δ p		0
abc		000

← - 1 W

Come per la SC > , per la condizione esterna 3 .

5.33. ISTRUZIONE SE 4

Salta sulla condizione esterna 4

Codice decimale : 0:
Codice binario : 0001 1000
Decod. parziali : Δ S. ΔQ. Δm. $\bar{a}$ $\bar{b}$ $\bar{c}$
Configurazione : XTbImIcIdIuTaFdFu

	0001	1000
Δ S	0001	
Δ Q	1	
Δ m		1
$\overline{abc}$		.000

← - 1 W

Come per la SC > per la condizione esterna 4 .

5.34. ISTRUZIONE STOP

Arresto

Codice decimale : 00
Codice binario : 0001 0001
Decod.parziali : $\overline{D}$, $\overline{B}$, $\overline{C}$, $\overline{d}$, $\overline{c}$, $\overline{b}$ = $\overline{STOP}$
Configurazione : X X X X X X X Fd Fu

← - 1 W

Arresta la preparazione ed esecuzione delle istruzioni.

Comprende la sola fase α ; nei primi due pdc si decodifica l'ordine STOP ; i successivi sette sono superflui.

Nel nono pdc si ripone α , ma non si dispone β . RC viene posto in R .

Al decimo μs di p9 , Cp passa a 01 e vi rimane, perchè l'assenza di segnali di fase mantiene RC nello stato R .

5.34.1. Preparazione STOP

Come la preparazione MA, con la differenza che al M8,5 di p9, β non viene disposto.

5.35. ISTRUZIONE +X

Moltiplicazione additiva

Codice decimale: 23
Codice binario : 0111 0110
Decod. parziali: ΔX ΔA Δp
Configurazione : LdLuImIcIdIuTaFdFu

	0111	0110
Δ X	01..	.110
Δ A	..11	....
Δ p	....	0...

Moltiplica un numero (Moltiplicando) proveniente da Memoria, a partire dalla posizione ImIcIdIu (modificata secondo Ta), di lunghezza LdLu (Lu modificato secondo Ta) per un numero (Moltiplicatore) in RT, nel registro TO, di lunghezza fissata dalla Marca.

Il moltiplicatore può essere segnato negativo (S_R disposto) o non segnato.

Il numero in Acc può essere segnato o non segnato: se è segnato, può essere normale o in complemento.

Se uno dei numeri è segnato, il risultato è segnato.

Il segno di Acc rimane comunque invariato: eventualmente si aziona il complementatore sul canale da Mem in base al prodotto dei segni (ottenuto in fase preparatoria), e/o si agisce (al termine dell'operazione) sul flip flop CN, secondo le regole esposte nella parte II^.

L'esecuzione della +X consta di un numero di sottofasi (dette cicli), pari al numero delle cifre di moltiplicatore, più uno se la cifra più significativa è $\geqslant 5$.

Ogni ciclo inizia con il trasferimento di una cifra dal Moltiplicatore (a partire dalla meno significativa) dal registro TO di RT al RAM . Nei successivi periodi si leggono via via le cifre del Moltiplicando, le si sommano a se stesse tante volte quante indicato dalla cifra di moltiplicatore nel RAM ; si somma infine il risultato con il contenuto di Acc, spostandosi di una cifra ad ogni ciclo. Il termine del ciclo è indicato dal segnale di fine somma φ', che riporta a 01 il Cp .

La fine dell'operazione è data dal segnale φ' dell'ultimo ciclo.

Se uno degli operandi da Mem o da Acc termina prima del segnale φ', si aziona il corrispondente generatore di zeri.

La marca viene riscritta in Acc. nel pdc in cui compare φ.

Ogni volta che si scrive in Acc una cifra che non dovrà essere più chiamata ai cicli successivi (1), si esamina l'uscita da UA : se è diversa da zero si ripone il flip flop IZ . Se al termine dell'esecuzione IZ è ancora disposto, il risultato è nullo (almeno limitatamente alla lunghezza LdLu).

Per limitare a 5 il numero massimo di somme con se stesse delle cifre da Mem, si complementa a 10 la cifra uscente da TO all'atto dell'immissione nel RAM, se è $\geqslant 5$: il risultato è corretto se si cambia il segno complessivo di operazione del ciclo in cui si complementa e si somma un'unità alla cifra di moltiplicatore del ciclo successivo.

5.35.1. Preparazione +X

Come per la preparazione MA ed inoltre:
al pdc 3 non si azzerano gli indicatori di Acc.
L'indicatore di zero IZ viene disposto al M5 pdc 9 .

5.35.2. Esecuzione +X

5.35.2.1. Periodo 1 delle prima sottofase

Si porta l'indirizzo del Moltiplicando (da Mem) da Raus O a W , e lo si copia di nuovo indietro in Raus O .

Si prepara l'indirizzo TO su V , scrivendo zero su Vd e contando +1 . L'indirizzo viene poi copiato indietro su Raus R., per il prossimo ciclo.

Si estrae il primo carattere del Md da Mem e lo si immette nello stat d'uscita Mem; la porta GMUA è bloccata fino a M9 da F02. Se è un segno, si dispone il flip flop S M4 , che rimane disposto per 4 pdc .

Si estrae la prima cifra da RT e la si immette nel RAM . Se tale cifra è $\geqslant 5$, la si complementa a 10 e si dispone d_{n-1} (che in ogni sottofase deve avvertire dell'avvenuta complementazione nella sottofase precedente). (2).

---

1) - Ciò avviene per tutti i pdc dell'ultima sottofase, ed inoltre per il pdc 2 di ogni sottofase, se Md non è segnato, nel pdc 3 di ogni sottofase se Md è segnato .-

2) - L'uscita da RT è inviata anche (inutilmente) in UA_2 . In UA_1 è presente zero bit oppure 5 (se D' è disposto). Le uscite di UA (normale e riporto) devono pertanto essere bloccate in p1 .

Si trasferisce il contenuto del cont. Rip (zero codice) nello Stat prod. SP (previamente cancellato), e si azzera Cont. R .

Si immette in UA_1 il contenuto dello Stat Mem, complementando o no, a 10 .

Si immette in UA_1 il contenuto di SP e si esegue la prima somma ciclica.

Si confronta Cp con RL .

M 0 - Rim a 01 Cp
- Canc W
- Canc V
- Rip UMZ (1)

M 1 - Disponi UMZ

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p 3 MO
- Int Raus ⟶ V : (2) è selzionato R da p2 M8,5
- Zero dec : scrive zero codice su Vd .

M 3 - Conta +1 V : scrive zero codice su Vu .

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di β

F 01 - Sua (1) - GAUA (1)

M 5,5 - Compara (1 g)

M 6 - Sonda Mem
- Sonda Acc
- RC in C
- Compl Acc : se $d_A = 1$ (d_{n-1} è riposto)
- +1 A : se $d_A = 1$

M 7 - GARAM
- Scritt W ⟶ Raus: è selezionato 0 da α p3 MO
- Scritt V ⟶ Raus: è selezionato R da α p2 M 8,5

M 7,5 - Rip d_{n-1} : superfluo in questa sottofase
- Pil Mem Scritt
- GMM
- Pil Acc Scritt
- GAA

---

1) - E' superfluo in questo pdc .-

2) - E' superfluo ma serve nelle sottofasi seguenti; non fa danno perchè R è stato interrogato (per cancellarlo) in α p 3 e subito dopo si è cancellato V .-

M 8 - Canc SP

M 8,5 - Disponi SM_4 : se da Mem è uscito un segno
- Cont R $\longrightarrow$ SP
- Disponi d_{n-1} : se $d_A = 1$

M 9 - Azzera Cont R

F02 - GMUA
- Compl Mem : se D è disposto
- +1 Mem : se D è disposto
- S u b : somma ciclica .

5.35.2.2. Periodo 2 della prima sottofase

Per UA_1 è sempre presente il contenuto dello Stat Mem .

Si eseguono le successive somme cicliche di tale contenuto fino ad un massimo di 4 . Gli eventuali riporti decimali fanno ogni volta avanzare di 1 il Cont R . Contemporaneamente se da Mem è uscito un segno in p1 , si trasferisce su W il contenuto di Raus O , indi si conta comunque -1 W ; infine (solo se da Mem in p1 uscì un segno), si copia indietro il contenuto di W su Raus O .-

Infine si porta l'indirizzo di partenza dell'Acc da Raus P a V e lo si copia poi di nuovo indietro in Raus P . Si estrae la seconda cifra del Md da Mem e la si immette nello stat d'uscita Mem : la porta GMUA è bloccata fino a M9 da F02 .

Si estrae la prima cifra di Acc e la si immette in UA_2 al termine delle somme cicliche iniziate nel pdc precedente: il risultato di tali somme è intanto immesso in UA_1 .

Se il primo carattere del M_d , estratto da Mem nel pdc precedente, fu un segno, si bloccano le uscite da UA e si rigenera Acc .

Se il primo carattere non fu un segno, si pone l'uscita da UA nella prima posizione di Acc : l'eventuale riporto decimale fa avanzare di 1 il Cont R (che poteva avere raggiunto la posizione 4 a causa dei riporti delle somme cicliche); si ripone IZ se si scrive in Acc una cifra significativa. Da ultimo si trasferisce comunque il contenuto di Cont R nello Stat Prod (previamente cancellato) e si azzera Cont R .

Si immette in UA_1 il contenuto di SP e si esegue la prima somma ciclica.

Si confronta Cp con RL .

M 0	- Sel Acc : il cessare di p1 consente la sel di Acc . - Sel Raus P - Conta +1 Cp - Rip UMZ : se nel pdc Precedente è uscita una cifra significativa da Mem - Canc V (1)
r SPO (2)	- Canc S.P.
UP 0,2	- GUA SP - +1 Cont R : se c'è riporto decimale
r SP 1	- Canc SP
UP 1,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 1,5	- Int Raus ⟶ V : è selezionato P da MO - Int Raus ⟶ W : solo se da Mem è uscito un segno (SM_4 disp)
r SP 2	- Canc SP
UP 2,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 3	- Conta -1 W
r SP 3	- Canc SP
UP 3,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 4,5	- Pil Mem Int - Pil Acc Int : è selezionato Acc da MO
M 5,5	- Compara
FO1	- S u a : somma il contenuto di Acc col risultato della somma ciclica - GAUA : se da Mem non è uscito un segno
M 6	- Sonda Mem - Sonda Acc
M 7	- Scritt W ⟶ Raus solo se da memoria è uscito un segno - Scritt V ⟶ Raus : è selezionato P da MO - GUAA : se da Mem in p1 non è uscito un segno - +1 Cont R se c'è riporto decimale - Rip IZ : se si scrive su Acc una cifra significativa.

---

1) - E' ottenuto da β p , Δ 3 ritardato di 1 μs .-

2) - Il numero di segnali e SP e UP è stabilito dalla cifra nel RAM .-

M 7,5	- Pil Mem Scritt - GMM - Pil Acc Scritt - GAA se da Mem è uscito un segno
M 8	- Canc SP
M 8,5	- Cont R ⟶ SP
M 9	- Azzera Cont R
F02	- GMUA - Compl Mem : se D e UMZ sono disposti - S u b : somma ciclica.

5.35.2.3. Periodo 3 della prima sottofase

In UA è sempre presente il contenuto dello Stat Mem . Si eseguono le successive somme cicliche di tale contenuto fino ad un massimo di quattro. Gli eventuali riporti decimali fanno ogni volta avanzare di 1 il Cont R .

Contemporaneamente si conta -1 W ; inoltre si trasferisce su V il contenuto di Raus P, si conta +1 V (quest'ultimo solo se nel pdc 1 da Mem non è uscito un segno), ed infine si copia indietro il contenuto di V su P . Si estrae la terza cifra del Md da Mem e la si pone nello stat d'uscita Mem : la porta GMUA è bloccata fino a M9 da F02 .

Se il carattere estratto da Mem nel 1° pdc fu un segno, si estrae di nuovo la prima cifra di Acc; se non fu un segno, si estrae la seconda cifra di Acc; si immette questa cifra da Acc in UA_2 al termine delle somme cicliche iniziate nel pdc precedente; il risultato di tali somme è intanto immesso in UA_1 .

Si pone l'uscita da UA nella prima posizione di Acc, se da Mem uscì un segno nel primo pdc, nella seconda posizione di Acc se da Mem non uscì un segno; l'eventuale riporto decimale fa avanzare di 1 il Cont R (che poteva aver raggiunto la posizione 4 a causa dei riporti delle somme cicliche); se si scrive in Acc una cifra significativa, si ripone IZ .

Si trasferisce il contenuto di Cont R nello Stat Prod (previamente cancellato) e si azzerano Cont R . Si immette in UA_1 il contenuto dello Stat Mem, complementando o no (a 9 se in uno dei precedenti è uscita da Mem una cifra significativa, a zero negli altri casi).

Si immette in UA_2 il contenuto di SP e si esegue la prima somma ciclica.

Si confronta Cp con RL .

M 0	- Conta +1 Cp - Rip UMZ : se nel pdc precedente è uscita da Mem una cifra significativa.
r s P 0	- Canc SP
UP 0,2	- GUASP - +1 Cont R : se c'è riporto decimale
r s P 1	- Canc SP
UP 1,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 1,5	- Int Raus ⟶ V (1) : è selezionato P da MO di p2 .
r s P 2	- Canc SP
UP 2,2	- GUASP - +1 Cont R se c'è riporto decimale
M 3	- Conta -1 W - Conta +1 V : se da Mem non uscì un segno nel pdc 1 .
r s P 3	- Canc SP
UP 3,2	- GUASP - Conta +1 Cont R : se c'è riporto decimale
r s P 4	- Canc SP
UP 4,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 4,5	- Pil Mem Int - Pil Acc Int : è sel Acc
M 5,5	- Compara
F 01	- S u a : somma Acc con risultato di somme cicliche - GAUA
M 6	- Sonda Mem - Sonda Acc
M 7	- GUAA - Rip IZ:se si scrive in Acc una cifra sifnificativa - Scritt V ⟶ Raus : è selezionato P da MO di p2 .

---

1) - Non disturba V , perchè il contenuto di V coincide con quello di Raus P ; ciò serve per cancellare Raus P .-

M 7,5	- Pil Mem Scritt - GMM - Pil Acc Scritt
M 8	- Canc SP
M 8,5	- Cont R ⟶ SP
M 9	- Azzera Cont R
F 02	- GMUA - Compl Mem : se D è disposto - +1 Mem : se D e UMZ sono disposti - S u b : somma ciclica.

5.35.2.4. Periodo 4 della prima sottofase

In UA_1 è sempre presente il contenuto dello Stat Mem . Si eseguono le successive somme cicliche di tale contenuto fino ad un massimo di quattro. Gli eventuali riporti decimali fanno avanzare di 1 in Cont R .

Contemporaneamente si conta -1 W ; inoltre, se da Mem uscì un segno in p1 , si trasferisce il contenuto di Raus P ; indi si conta comunque +1 V, ed infine, (solo se da Mem uscì un segno) si copia indietro il contenuto di V su Raus P .

Si estrae la successiva cifra del Md da Mem e lo ripone nello stat d'uscita Mem : la porta GMUA è bloccata fino a M9 da F02 .

Si estrae la successiva cifra di Acc e la si immette in UA_2 al termine delle somme cicliche iniziate nel pdc precedente; il risultato di tali somme è intanto immesso in UA_1 .

Si pone l'uscita da UA nella successiva posizione di Acc; l'eventuale riporto decimale fa avanzare di 1 il Cont R (che poteva aver raggiunto la posizione 4 a causa dei riporti delle somme cicliche); se si scrive in Acc una cifra significativa, si ripone IZ .

Si trasferisce il contenuto di cont R nello stat Prod (previamente cancellato) e si azzera cont R .

Si immette in UA_1 il contenuto dello stat Mem , complementando o no (a 9 se in uno dei pdc precedenti è uscita da Mem una cifra significativa, a 10 negli altri casi).

Si immette in UA_2 il contenuto di SP e si esegue la prima somma ciclica.

Si confronta Cp con RL .

Comandi come in p3 salvo che ad M3 si conta comunque +1 V , mentre Int Raus ⟶ V a M1,5 e Scritt V ⟶ Raus a M7 si hanno solo se da Mem in p1 uscì un segno. Inoltre a M8,5 si ripone SM_4 .

5.35.2.5. Periodo 5 della prima sottofase

Come per il p4 , salvo che non si hanno i comandi Int Raus ⟶ V, Rip SM_4 .

I successivi periodi sono tutti uguali a p5, fino al pdc in cui ha inizio la procedura di fine di sottofase che è identica (come casistica) a quella della +MA .

La struttura dei pdc finali può essere ricavata dalla struttura del pdc 5 della +X, prima sottofase.

5.35.2.6. Periodo 1 della seconda sottofase

Si porta l'indirizzo del primo carattere diverso dal segno del Md (da Mem) da Raus 0 (1) a W , e lo si copia poi di nuovo indietro in Raus 0 .

Si prepara l'indirizzo da chiamare in RT, trasferendo su V il contenuto di Raus R e contando +1 V : tale indirizzo viene poi copiato indietro su Raus R . Si estrae la prima cifra significativa del Md da Mem e la si immette nello stat d'uscita Mem ; la porta GMUA è bloccata fino a M9 da F02 .

Si estrae la seconda cifra da RT e la si immette nel RAM, sommando +1 se nella precedente sottofase si era disposto d_{n-1} . Se la cifra da RT è $\geqslant$ 5, la si complementa (a 10 se d_{n-1} è riposto, a 9 se è disposto), (2), e si dispone d_{n-1} (dopo averlo comunque riposto).

Si trasferisce il contenuto dello Stat Mem, complementando o no a 10 .

Si immette in UA_2 il contenuto di SP e si esegue la prima somma ciclica.

---

1) - Date le operazioni eseguite nei pdc 1 e 2 della prima sottofase, Raus 0 contiene l'indirizzo del 1° carattere diverso dal segno del M α .

2) - Vedi nota (2) pag. 159 .

Si confronta Cp con RL .

M 0 - Rim A 01 Cp
- Canc W
- Canc V
- Rip UMZ (1)

M 1 - Disp UMZ

M 1,5 - Int Raus ⟶ W : è selezionato 0 da α p3 M0
- Int Raus ⟶ V : è selezionato R da 8,5 del pdc finale della precedente sottofase.
- Zero dec (2)

M 3 - Conta +1 V

M 4,5 - Pil Mem Int
- Pil Acc Int : è selezionato T dall'inizio di β .

F 01 - S u a (1)
- GAUA (1)

M 5,5 - Compara

M 6 - Sonda Mem
- Sonda Acc
- RC in C
- Compl Acc : se $d_A = 1$
- +1 Acc : se $d_A = 1$ e d_{n-1} è riposto, ovvero se $d_A = 0$ e d_{n-1} è disposto.

M 7 - GARAM
- Scritt W ⟶ Raus : è selezionato 0 da α p3 M0
- Scritt V ⟶ Raus : è selezionato R da α p2 M8,5

M 7,5 - Rip d_{n-1}
- Pil Mem Scritt
- GMM
- Pil Acc Scritt
- GAA

M 8 - Canc SP

M 8,5 - Disp UMZ
- Cont R ⟶ SP
- Disp d_{n-1} : se $d_A = 1$

---

1) - E' superfluo in questo pdc .-
2) - E' superfluo, ma non dannoso perchè su V dec è scritto comunque zero codice.-

M 9 - Azzera Cont R

F 02 - GMUA
- Compl Mem : se D è disposto
- S u b : somme cicliche.

5.25.2.7. Periodo 2 della seconda sottofase

Su UA_1 è sempre presente il contenuto dello stat Mem . Si eseguono le successive somme cicliche di tale contenuto, fino ad un massimo di 4 . Gli eventuali riporti decimali fanno ogni volta avanzare di 1 il Cont R .

Contemporaneamente si conta +1 W, ed inoltre si trasferisce su V il contenuto di Raus P, che è l'indirizzo di partenza dell'acc nella precedente sottofase, aumentato di 1) e lo si copia di nuovo indietro in Raus P .

Si estrae la seconda cifra del Md da Mem e la si immette nello stat d'uscita Mem : la parte GMUA è bloccata fino a M9 da F02 .

Si estrae la seconda cifra di Acc e la si immette in UA_2 al termine delle somme cicliche iniziate nel pdc precedente: il risultato di tali somme è intanto immesso in UA_1 . Si pone l'uscita da UA nella seconda posizione di Acc : l'eventuale riporto decimale fa avanzare di 1 il Cont R (che poteva aver raggiunto la posizione 4 a causa dei riporti delle somme cicliche). Si ripone IZ se si scrive in Acc una cifra significativa.

Si trasferisce il contenuto di cont R nello SP (previamente cancellato) e si azzera Cont R . Si immette in UA_1 il contenuto dello Stat Mem complementando o no (a 10 se nel pdc precedente è uscito zero, a 9 se è uscita una cifra significativa).

Si immette in UA_2 il contenuto di SP e si esegue la prima somma ciclica.

Si confronta Cp con RL .

M 0
- Sel Acc
- Sel Raus P
- Conta +1 Cp
- Rip UMZ : se nel pdc precedente è usata una cifra significativa da Mem
- Canc V

r s PO - Canc SP

UP 0,2
- GUASP
- +1 Cont R : se c'è riporto decimale

r s P1	- Canc SP
UP 1,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 1,5	- Int Raus →V : è selezionato P da MO
r s P2	- Canc SP
UP 2,2	- GUASP - +1 Cont R : se c'è riporto decimale
M 3	- Conta -1 W
r s P3	- Canc SP
UP 3,2	- +1 Cont R : se c'è riporto decimale
r s P4	- Canc SP
UP 4,2	- GUASP - +1 Cont R : se c'è riporto
M 4,5	- Pil Mem Int - Pil Acc Int
M 5,5	- Compara
F 01	- S u a : somma Acc col risultato delle somme cicliche - GAUA
M 6	- Sonda Mem - Sonda Acc
M 7	- Scritt V → Raus : è selezionato P da MO - GUAA - +1 Cont R : se c'è riporto - Rip IZ : se si scrive in Acc una cifra significativa.
M 7,5	- Pil Mem Scritt - GMM - Pil Acc Scritt
M 8	- Canc SP
M 8,5	- Cont R → SP
M 9	- Azzera Cont R
F 02	- GMUA - Compl Mem : se D è disposto - +1 Mem : se D è disposto - S u b : somma ciclica.

5.35.2.8. Periodo 3 della seconda sottofase

Come per il pd 2 riferito alla 3^ cifra da Mem ed inoltre si conta +1 V a M3 , per cui si estrae la 3^ cifra di Acc .

5.35.2.9. Periodo 4 della seconda sottofase

Come per il pdc 3, senza trasferimento da P a V e viceversa.

I successivi periodi sono tutti uguali a p4 , fino al pdc in cui ha inizio la procedura di fine, di sottofase che è identica (come casistica) a quella di fine + MA .

La struttura dei pdc finali può essere ricavata dalla struttura dei pdc finali +MA, integrata dalla struttura del pdc 4 , 2^ sottofase, +X .

Le successive sottofasi sono tutte uguali alla seconda, fino a quella in cui si giunge all'ultima cifra del M r : nel pdc 1 compare allora un segnale f_A che dispone il flip flop $\emptyset_R$. Se d_{n-1} non viene disposto nel corso del 1° pdc, questa sottofase è l'ultima.

Se d_{n-1} viene disposto, si esegue ancora una ulteriore sottofase durante la quale la cifra di moltiplicatore è zero e viene simulata dal generatore di zeri sul canale di accumulatore.

5.36. ISTRUZIONI RELATIVE ALLE OPERAZIONI SU NASTRO MAGNETICO

Generalità

Sono composte di una fase α di preparazione ed, in generale, di due fasi β e γ di esecuzione.

Nel corso della fase α vengono prelevati dalla memoria i caratteri di funzione che, decodificati, forniscono al G.U.N. segnali caratteristici delle varie operazioni (Δ LN : lettura di nastro verso Memoria; Δ RN registrazioni a nastro da memoria; Δ TN trascrizione tra nastri; Δ DUB disponi unità e blocco; J, indietro).

Vengono inoltre inviati al GUN indicazioni relative alle unità a nastro da operare e, sulle istruzioni tipo TN e sulla DUB , relative al numero di blocchi da trascrivere o rispettivamente, da contare.

Una operazione a nastro può iniziare solamente se il nastro è inizialmente fermo.

Nel caso in cui il nastro sia fermo il GUN invia alla macchina il segnale ØK è allora consentito nel caso di operazioni magnetiche il passaggio alla fase β al termine di α . Se è invece presente il segnale $\overline{ØK}$, il GUN non è disposto ad eseguire la istruzione, non è consentito il passaggio alla fase β e viene ripetuta la fase α fino a che non viene inviato il segnale ØK .

Il tempo necessario per leggere dal nastro o per registrare sul nastro un carattere, periodo di nastro, è di 100 μs , mentre il periodo di macchina è di 10 μs .

In registr. occorre quindi che i car. siano prelev. da mem e forniti al GUN ogni 100 μs ; per questo viene inviata al GUN la forma d'onda F 15 (durata dal M 1 al M 5) che si aziona un divisore per 10 . Ogni 10 impulsi F 15 nasce una segnale S_2 che viene inviato alla macchina ove permette la nascita del segnale p_n in corrispondenza del quale avviene il trasferimento da memoria al GUN .

Non è possibile iniziare la registrazione prima che il nastro abbia raggiunto la velocità di regime : per questo il GUN è dotato di organi che permettono la nascita del primo S_2 solo quando tale condizione è verificata. Sia ha quindi un intervallo di attesa prima della registrazione del primo carattere.

In lettura il trigger S_2 viene invece inviato dal GUN alla macchina in seguito alla lettura del carattere sul nastro.

Nel tempo intercorrente tra un p_n ed il successivo non avvengono operazioni in macchina.

Il trasferimento da nastro a memoria o da memoria a nastro avviene nel corso della fase γ .

Le operazioni eseguite nella fase β sono diverse da istruzione a istruzione.

5.37. ISTRUZIONE RNa

Codice decimale 90
Codice binario 1100 0001
Decod. parziali Δ N Δ E Δ I Δ P Δ RN
Configurazione XnIIIITFF

	1100	0001
Δ N	11..	....
Δ E	1...	....
Δ I	..00	....
Δ P	....	0...
Δ RN	110.	

Registra dalla memoria all'unità a nastro n (mod. T), procedendo in avanti. Il trasferimento ha inizio dalla posizione IIII della Mem. (mod T) e procede fino al segnale di fine blocco . Ciò è ottenuto nel modo seguente.

Nel corso della fase di preparazione, che si svolge analogamente alla preparazione MA , vengono inviati al GUN il segnale LN e l'indicazione dell'unità a nastro da selezionare.

Segue una fase β di 3 pdc che è necessaria per esigenze interne del GUN : esistono infatti istruzioni in cui il dato estratto da Mem in p9 è utilizzato dal GUN in p1 ; in p2 il GUN utilizza la f.d.O. F 15 per generare un segnale Q che ripone il F.F. ØH ; si prolunga la fase β ancora di un pdc per evitare che la comparsa di γ mentre ancora non è scomparso ØH generi tempestivamente un segnale φ .

Si ha, successivamente, una fase γ che si inizia con un intervallo di attesa fino all'arrivo del primo S_2 che dispone il F.F. TN che genera p_n e permette il pilotaggio della memoria e l'estrazione del primo carattere che viene inviato al GUN .

Tra un p_n ed il successivo non si eseguono operazioni in macchina. La fine dell'operazione si ha quando dal GUN viene decodificato il segnale di fine blocco ($ £) che genera ØH è, di

conseguenza, il segnale di fine φ .

5.37.1. Preparazione RNa - Fase α

Periodi 1-7 come per la preparazione MA, inoltre:
- in p2 a M 8,5 se è presente ØK , si ripone AN
- in p5 a M 8,5 se è presente ØK , si dispone AN

Periodo 8 : come per la preparazione MA, salvo che il carattere n viene inviato al GUN invece di essere registrato in RL unità .

Periodo 9 : come per la preparazione MA; il carattere riempitivo X viene registrato in RL dec. Si può passare alla successiva fase β solo se è presente $\overline{AN}$, altrimenti si ripone RC , il Cp ritorna a 01 a MO , si inibisce +10 W ; si conta invece +1 W , e si ripete la fase α fino a che non compare ØK .

5.37.2. Esecuzione RNa - Fase β

5.37.2.1. Periodo 1

Si trasferisce l'indirizzo dell'operando da Raus O a W

M O - Rip a 01 Cp
- Canc W (1)
- Canc V (1)
- Sel Raus P (1)

M 1,5 - Interrog. Raus $\longrightarrow$ W (è sel. O da α p3)

M 5,5 - Compara (1)

M 6 - RC in C

5.37.2.2. Periodo 2

M O - Conta +1 Cp

M 5,5 - Compara (1)

---

1) - E' superfluo.-

5.37.2.3. Periodo 3

Si ripone β , si dispone γ

M 0 - Conta +1 Cp

M 5,5 - Compara (1)

M 6 - Disp. γ

M 8 - Rip β
- RC in R

5.37.3. Fase γ

Intervallo di attesa
Arriva S_2 , a M 9 nasce p_n

5.37.3.1. Periodo 1

Si pilota la Mem, si invia al GUN il 1° carattere estratto, si rigenera lo stesso carattere in Mem. Si ripone p_n

M 0 - Rip a 01 Cp

M 4,5 - Pil Mem Int

M 5,5 - Compara (1)

M 6 - Sonda Mem
- Mem $\longrightarrow$ GUN
- RC in C

M 8,5 - Pil Mem Scritt
- GMM
- Rip p_n

Intervallo di attesa
Arriva S_2 , a M 9 nasce p_n

5.37.3.2. Periodo 2

Si estrae il secondo carattere da Mem, lo si invia al GUN e lo si rigenera in Mem. Si ripone p_n

---

1) - E' superfluo.-

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara (1)

M 6 - Sonda Mem
- Mem ⟶ GUN

M 8,5 - Pil Mem Scritt
- GMM
- Rip p_n

I successivi periodi sono tutti eguali a p2 fino a che il GUN e lo si rigenera in Mem.

Il GUN decodifica il fine blocco e invia il segnale ØK . Nasce φ , si ripone γ , si ripone p_n , si dispone α , si pone RC in R .

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara (1)

M 6 - Sonda Mem
- Mem ⟶ GUN

M 8 - RC in R

M 8,5 - Pil Mem Scritt
- GMM
- Rip p_n
- Rip γ

M 9 - Disp. α

---

1) - E' superfluo.-

5.38. ISTRUZIONE RNi

Codice decimale 95
Codice binario 1100 1001
Decod. parziali ΔN ΔE Δm ΔI Δ RN
Configurazione XnIIIITFF

	1100	1001
Δ N	11..	....
Δ E	1...	....
Δ m	....	1...
Δ I	..00	....
Δ RN	110.	....

Registra dalla memoria all'unità a nastro n (mod T), procedendo indietro. Il trasferimento ha inizio dalla posizione di mem. IIII (mod T) e prosegue fino al segnale di fine blocco. Modalità come per la RNa . Durante la fase α viene inviato al GUN anche il decod. J .

5.38.1. Preparazione RNi

Come per la preparazione RNa .

5.38.2. Esecuzione RNi

Come per l'esecuzione RNa salvo che si ha sempre -1 W in luogo di +1 W .

5.39. ISTRUZIONE LNa

Codice decimale 70
Codice binario 1111 0001
Decod. parziali Δ N ΔE ΔA ΔP ΔLN
Configurazione

	1111	0001
Δ N	11..	....
Δ E	1...	....
Δ A	..11	....
Δ P	....	0...
ΔLN	1111	....

Legge dall'unità a nastro n alla memoria procedendo in avanti; il trasferimento ha inizio a partire dalla posizione ii del blocco espressa in decine e all'indirizzo II della memoria espresso in centinaia. Il trasferimento riguarda L decine di caratteri. Se L = 0 il trasferimento procede fino al segnale di fine blocco. n, II, ii sono modificabili con T .

Nel corso della fase α , che si svolge come la preparazione RNa, vengono registrati nell'ordine in Raus 0 .

Nella successiva fase β , si trasferisce il contenuto di Raus 0 in W , si ha una serie di periodi di attesa (in cui si ha la conta +1 Cp) fino a che il GUN comincia a leggere i caratteri del blocco che devono essere contati e non inviati alla macchina, ogni dieci caratteri letti si ha il segnale DO che provoca la conta -1 W . Il segnale DO è generato circa 30 μs prima di S_2 e dura 100 μs , esso è quindi presente in macchina assieme a p_n .

Le temporizzazioni del segnale DO, di S_2 , del F.F. TN , e di p_n sono indicate in fig. (1).

Quando sono state contate ii decine di caratteri è verificata la condizione $W_{10} = W_1 = 0$, si ripone β , si dispone γ . Se ii = 00 la fase β termina in p3 come nella RNa .

Si ha infine una fase γ in cui vengono inviati in corrispondenza della lettura di ogni carattere, i segnali S_2 dal GUN alla macchina, nascono di conseguenza i segnali p_n ed avvengono trasferimenti da GUN a Mem .

Se L ≠ 0 la fine dell'operazione si ha quando la comparazione dà esito favorevole; nasce allora ØC che genera il segnale di fine φ .

Se L = 0 la fine dell'operazione si ha quando il GUN decodifica il fine blocco; nasce allora $Ø_H$ che genera il segnale di fine φ .

5.39.1. Preparazione LNa

Come per la preparazione RNa . Inoltre in p3 M 2 si scrive zero in RL unità.

5.39.2. Esecuzione LNa - Fase β

5.39.2.1. Periodo 1

Come per β p1 della RNa : inoltre si scrive zero cod. in RL unità.

5.39.2.2. Periodi 2 e successivi

Come per β p2 della RNa .
Ad un certo istante nasce DO, indi S_2 e p_n .
Si ha allora un pdc diverso dai precedenti.

5.39.2.3. Periodo m

Si esegue una conta di -1 W .

M 0 - Conta +1 Cp

M 3 - Conta -1 W

M 5,5 - Compara (1)

I successivi periodi sono tutti eguali a p2 fino a che, dopo 10 p_n , è presente un secondo DO .
Si ha allora un pdc eguale all'm-esimo.
Quando $W_{10} = W_1 = 0$ si ha l'ultimo periodo.

5.39.2.4. Ultimo periodo

Conta -1 W e si raggiunge la configurazione $W_{10} = W_1 = 0$.
Si ripone β , si dispone γ , si pone RC in R .

---

1) - E' superfluo.-

M 0 - Conta +1 Cp

M 3 - Conta -1 W

M 5,5 - Compara (1)

M 6 - Disp.

M 8 - Rip
- RC in R

M 8,5 - Rip. p_n

Si osserva che comunque la fine non si può avere se il contatore principale si trova nelle configurazioni 01 o 02 . In tal caso si ha il passaggio alla fase γ rispettivamente con 20 e con 10 μs di ritardo.

5.39.3. Fase γ

Si attende l'arrivo di S_2 che permette la nascita di p_n a M 9 .

5.39.3.1. Periodo 1

Si pilota la Mem. senza sonda, si confronta Cp con RL, si trasferisce in mem. il carattere inviato dal GUN .

M 0 - Rip a 01 Cp

M 4,5 - Pil Mem Int

M 5,5 - Compara

M 6 - RC in C

M 7 - GUN $\longrightarrow$ Mem

M 8,5 - Pil Mem Scritt
- Rip p_n

Si attende l'arrivo di S_2 che permette la nascita di p_n a M 9 .

5.39.3.2. Periodo 2

Si pilota la mem senza sonda, si trasferisce in mem., il successivo carattere inviato dal GUN, si confronta Cp con RL .

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara

M 7 - GUN ⟶ Mem

M 8,5 - Pil Mem Scritt
- Rip. p_n

I successivi periodi sono tutti eguali a p2 . Se il carattere L dell'istruzione è $\neq$ 0 si ha la seguente procedura di fine.

5.39.3.3.1. Ultimo periodo (caso L $\neq$ 0)

Si pilota la mem. senza sonda, si trasferisce in Mem. il carattere inviato dal nastro, la comparazione dà esito favorevole, nasce $\emptyset_c$, indi φ ; si ripone γ , si dispone α , si pone RC in R .

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara
- Nasce $\emptyset_c$
- Nasce φ

M 7 - GUN ⟶ Mem

M 8 - RC in R

M 8,5 - Pil Mem Scritt
- Rip γ
- Rip. p_n

M 9 - Disp. α

Se L = 0 il trasferimento ha luogo fino al segnale di fine blocco. Si ha allora la seguente procedura di fine.

5.39.3.3.2. Ultimo periodo (caso L = 0)

Si pilota la mem. senza sonda, si trasferisce in Mem. il carattere inviato dal GUN . Il GUN decodifica il segnale di fine blocco e invia ØH , nasce φ , si ripone γ , si pone RC in R, si dispone α .

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara

M 7 - GUN $\longrightarrow$ Mem

M 8 - RC in R

M 8,5 - Pil Mem Scritt
- Rip γ
- Rip pn

M 9 - Disp. α

5.40. ISTRUZIONE LNi

Codice decimale 75
Codice binario 1111 1001
Decod. parziali Δ N. ΔE. Δ m. Δ A
Configurazione LnIIiiTFF

	1111	1001
Δ N	11..	....
Δ E	1...	....
Δ m	....	1...
Δ A	..11	....
Δ LN	11.1	....

Legge dall'unità a nastro n procedendo indietro: il trasferimento ha inizio a partire dalla posizione ii del blocco espresso in decine e all'indietro II della Mem. espresso in centinaia. Il trasferimento riguarda L decine di caratteri. Se L = 0 il trasferimento prosegue fino al segnale di fine blocco. n, II , ii sono modificabili con T .

Modalità come per la LNi . Durante la fase ∝ viene inviato al GUN anche il segnale J .

5.40.1. Preparazione LNi

Come per la preparazione LNa .

5.40.2. Esecuzione LNi

Come per l'esecuzione LNa, salvo che si ha sempre -1 W anzichè +1 W .

5.41. ISTRUZIONE TN

Codice decimale 80
Codice binario 1110 0001
Decod. parziali Δ N Δ E Δ P Δ T
Configurazione $n_1 n_2$IIIITFF

	1110	0001
Δ N	11..	....
Δ E	1...	....
Δ P	....	0...
Δ T	..10	....

Trascrive dall'unità a nastro n_1 , all'unità a nastro n_2 un numero IIII di blocchi, la trascrizione avviene con le due unità funzionanti in avanti (n_2 IIII mod T).

Durante la fase α si inviano al GUN, il decodificato Δ TN, la indicazione del numero di blocchi da trasferire e delle unità a nastro da operare.:

Segue una fase β di 2 pdc . Durante tale fase non avvengono operazioni in macchina, in β p1 il GUN utilizza il carattere uscito da mem. in α p9 , in β p2 nasce il segnale Q_β che ripone il F.F. ØH .

Al termine di β viene disposto α , si può quindi passare ad un'altra istruzione senza attendere l'esecuzione della TN .

5.41.1. Preparazione TN

5.41.1.1. Periodi 1-3 come per la preparazione MA .

5.41.1.2. Periodi 4-9 come per la preparazione MA , con le differenze seguenti:

- I caratteri IIII oltre ad essere registrati inutilmente in Raus 0, vengono inviati al GUN .
- Il carattere n_2 viene inviato al GUN
- Il carattere n_1 viene inviato al GUN ed è registrato inutilmente in RL unità.

Modalità di fine preparazione come per la RNa .

5.41.2. Fase β

5.41.2.1. Periodo 1

M 0 - Rip a 01 Cp
- Canc. W (1)
- Canc. V (1)
- Sel P (1)

M 1,5 - Interrog. Raus 0 ⟶ W (1)

M 5,5 - Compara (1)

M 6 - RC in C .

5.41.2.2. Periodo 2

M 0 - Conta +1 Cp

M 5,5 - Compara (1)

M 8 - Rip β
- RC in R .

M 9 - Disp α .

---

1) - E' superfluo.-

5.42. ISTRUZIONE DUB

Codice decimale 85
Codice binario 1110 1001
Decod. parziali Δ N Δ E Δm ΔT
Configurazione

	1110	1001
Δ N	11..	....
Δ E	1...	....
Δ m	....	1...
Δ T	..10	....

Fa procedere l'unità n di IIII (n e IIII mod. T) blocchi in avanti o indietro a seconda che $J < 5$ o $J \geq 5$.

Nel corso della fase α che si svolge analogamente alla preparazione TN , vengono inviati al GUN il decodificato Δ DN , eventualmente J , e le indicazioni del numero di blocchi da contare e dell'unità a nastro da operare.

Segue una fase β di 2 pdc al termine della quale viene disposto α ; si può quindi passare alla successiva istruzione senza attendere l'esecuzione della DUB .

5.42.1. Preparazione DUB

Come per la preparazione TN ; il carattere J viene decodificato in macchina e origina eventualmente il segnale di "indietro" J che viene inviato al GUN .

5.42.2. Esecuzione DUB

Come per l'esecuzione TN .

5.43. ISTRUZIONE NDNa

Codice decimale $ 0
Codice binario 1101 0001
Decod. parziali Δ N Δ E Δ P Δ RN Δ LN
Configurazione $n_1 n_2$IIIITFF

	1101	0001
Δ N	11..	....
Δ LN	11.1	....
Δ E	1...	....
Δ RN	110.	....
Δ P	....	0...

Registra dalla memoria procedendo in avanti all'unità a nastro n_2 e simultaneamente legge dall'unità a nastro n_1 verso memoria.

Ciascun carattere letto va ad occupare via via la posizione da cui si è prelevato il carattere da registrare.

Le informazioni vengono lette e registrate a gruppi di caratteri compresi tra particolari bicaratteri di servizio intercalati nel blocco (£ £). Il prelevamento procede secondo l'ordine stabilito da una direttrice contenuta in memoria a partire dall'indirizzo IIII .

Tale direttrice contiene gli indirizzi del primo carattere di ogni gruppo. L'ultima parola contenuta nella direttrice è l'indirizzo del carattere p2 dell'istruzione NDNa , n_2 e IIII sono modificabili con T .

L'operazione procede fino al segnale di fine blocco.

La fase α di preparazione è analoga alla preparazione MA ; vengono inviati al GUN : decodificati Δ LN Δ RN e le indicazioni delle unità a nastro da operare.

Segue una fase β di esecuzione uguale alla fase β della RNa, inoltre in p3 M 6 si dispone u .

Si ha infine una fase γ cui si sovrappongono a tratti fasi μ .

All'inizio di γ è presente anche μ .

La fase μ ha la durata di 8 p.d.c.

Nel corso di μ si cancella il Raus I, si trasferisce in W l'indirizzo di partenza della direttrice, immagazzinando in Raus O nel corso di α (ciò vale per la prima fase μ per la successiva l'indirizzo della direttrice è stato scritto in Raus O al termine della μ precedente), e si interroga la memoria trasfe-

rendo in Raus I la prima parola contenuta nella direttrice, infine si ricopia in Raus 0 il contenuto di W (dopo che si è contato -1 W 4 volte) e cioè l'indirizzo di partenza della successiva parola della direttrice. Si ripone μ . Si trasferisce il contenuto di Raus I (indirizzo del primo carattere del gruppo) in W . Si attende l'arrivo di S_2 trasferimento del primo carattere letto da n , a mem. Si prosegue il trasferimento ad ogni p_n fino a che non viene decodificato il fine gruppo (£ £).

La presenza del decodificato σ = £ £ origina una nuova fase μ nel corso della quale viene prelevato dalla direttrice l'indirizzo di partenza del secondo gruppo da trasferire . Successivamente, finita μ , avviene il trasferimento tra la memoria e l'unità a nastro n_1 e n_2 .

Si osserva che, siccome la fase μ dura 8 p.d.c., essa risulta compresa tra due p_n per cui al p_n successivo a quello in cui si è letta la seconda £ , che ha dato origine alla fase μ , si può iniziare il trasferimento del primo carattere del gruppo successivo. Il ciclo di operazioni descritto continua fino alla comparsa del fine blocco.

La fine dell'operazione avviene al termine dell'ultima fase μ, dopo la decodificazione del fine blocco; nasce φ , si dispone α , si ripongono γ e μ .

Durante l'ultima fase μ viene trasferito in Raus I l'indirizzo del carattere p2 dell'istruzione NDNa ed è così reso possibile il passaggio all'istruzione successiva.

5.43.1. Preparazione NDNa

Come per la preparazione RNa .
Inoltre il carattere n_1 viene utilizzato dal GUN .

5.43.2. Esecuzione NDNa

Fase

Come per la RNa . Inoltre a M 6 di p3 di dispone μ .

5.43.3. Fase γ e μ (fig. 2)

5.43.3.1. Periodo 1 (2)

Si rip. a 01 Cp, si cancella il Raus I

M 0 - Rip a 01 Cp
- Sel I
- Canc. I

M 1,5 - Int Raus ⟶ W (1)

M 5,5 - Compara (1)

M 6 - RC in C

5.43.3.2. Periodo 2 (2)

Si seleziona 0, si canc. W

M 0 - Conta +1 Cp

M 3 - Sel Raus 0

M 5 - Canc. W

M 5,5 - Compara (1)

5.43.3.3. Periodo 3 (2)

Si trasferisce in W il contenuto di Raus 0 .

M 0 - Conta +1 Cp

M 1,5 - Interrog. Raus ⟶ W (è selz. 0 da p2 M 3)

M 5,5 - Compara (1)

5.43.3.4. Periodo 4

Si pilota la memoria, si estrae da mem il primo carattere della direttrice e lo si pone in Raus I unità transitando attraverso

---

1) - E' superfluo.-

2) - Nei pdc 1.2.3.8 viene anche pilotata la memoria inutilmente; si rigenera il contenuto delle posizioni interrogate.-

UA e sommando +O .

M 0 - Conta +1 Cp

M 3 - Sel Raus I

M 4,5 - Pil Mem Int

M 5,5 - Compara (1)

M 6 - Sonda Mem.

F 01 - GMUA
- Zero Acc

M 7 - GUAG
- UA ⟶ Raus I

M 8,5 - Pil Mem Scritt
- GMM

5.43.3.5. Periodo 5

Si estrae da memoria il successivo carattere e lo si pone in Raus I decine transitando attraverso UA e sommando + O.

M 0 - Conta +1 Cp

M 3 - Conta -1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara

M 6 - Sonda Mem

F 01 - GMUA
- UA ⟶ Raus I

M 8,5 - Pil Mem Scritt
- GMM

5.43.3.6. Periodo 6

Come per il periodo 5 ma non con riferimento alle centinaia di I .

---

1) - E' superfluo.-

5.43.3.7. Periodo 7

Come per il periodo 5 ma con riferimento alle migliaia di I .

5.43.3.8. Periodo 8 (2)

Si seleziona Raus 0, si conta -1 W si trasferisce in 0 il contenuto di W , si pone RC in R, si ripone μ .

M 0 - Conta +1 Cp

M 3 - Sel Raus 0
- Conta -1 W

M 5,5 - Compara (1)

M 7 - Scritt W $\longrightarrow$ Raus (è selez. 0 da M 3)

M 8 - RC in R

M 8,5 - Rip. μ

Al successivo M 0 rip. a 01 Cp , Sel I
" " M1,5 Interrog. Raus $\longrightarrow$ W (è selez. I da M 0).
Si attende l'arrivo di S_2 che permette la nascita di p_n .

5.43.4.1. Periodo 1

E' nato p_n . Si pilota la memoria in interrogaz., si estrae il primo carattere del gruppo e lo si invia al GUN, si pilota la mem. in scrittura senza aprire GMM, si scrive in mem. il carattere in arrivo dal GUN .

M 4,5 - Pil Mem Interrog.

M 6 - Sonda Mem
- Mem $\longrightarrow$ GUN
- RC in C

---

1) - E' superfluo.-

2) - Nei pdc 1-2-3-8 viene anche pilotata la memoria inutilmente; si rigenera il contenuto delle posizioni interrogate.-

M 7 - GUN ⟶ Mem

M 8,5 - Pil Mem Scritt
- Rip p_n

Si attende l'arrivo del successivo S_2 .

5.43.4.2. Periodo 2

Si pilota la mem. in interrog., si estrae il secondo carattere del gruppo e lo si invia al GUN, si pilota la Mem in scritt. senza aprire GMM, si scrive in mem il carattere in arrivo dal GUN .

M 0 - Conta +1 Cp

M 3 - Conta +1 W

M 4,5 - Pil Mem Int

M 5,5 - Compara

M 6 - Sonda Mem
- Mem ⟶ GUN

M 7 - GUN ⟶ Mem

M 8,5 - Pil Mem Scritt
- Rip p_n

I successivi periodi sono tutti eguali a p_2 .

5.43.4.3. Durante il pdc m viene letto il primo carattere £ che compone il bicarattere £ £ di fine gruppo.

5.43.4.4. Periodo m

Come per il pdc 2 , inoltre il carattere £ , presente da M6 fino al successivo M5 , viene decodificato e dispone, a M4 , il F.F. £ .

5.43.4.5. Periodo m+1 (fig. 3)

Come per il pdc 2 , inoltre viene letto e decodificato il secondo carattere £ e, a M 8, nasce u e si pone RC in R .

5.43.5. Fase γ e μ

5.43.5.1. Periodi 1-8

Come per la fase γ e μ già descritta.

Successivamente si ha una nuova fase γ in cui si operano i trasferimenti, poi una fase γ e μ e così via.

L'operazione termina con una fase γ e μ quando sia stato in precedenza decodificato dal GUN il segnale di fine blocco.

5.43.6. Ultima fase γ e μ

5.43.6.1. Periodi 1-7

Come nella fase γ e μ già descritta.

5.43.6.2. Periodo 8

Come nella fase γ e μ già descritta; inoltre a M 0 nasce φ , a M 8 si pone RC in R , a M 8,5 si ripongono γ e μ , a M 9 si dispone α .

5.44. I S T R U Z I O N E NDNi

Codice decimale	$ 5
Codice binario	1101 1001
Decod. parziali	Δ N Δ E Δ m
Configurazione	$n_1 n_2$IIIITFF

	1101	1001
Δ N	11..	....
Δ E	1...	....
Δ m	....	1...

Come per la NDNa ma procedendo indietro.
Durante la fase α viene inviato al GUN anche il segnale J .

5.44.1. Preparazione NDNi

Come per la preparazione NDNa .

5.44.2. Esecuzione NDNi

Come per l'esecuzione NDNa salvo che in p_n si ha sempre -1 W anzichè +1 W .